Part 2

Elements of Bayesian Decision Theory

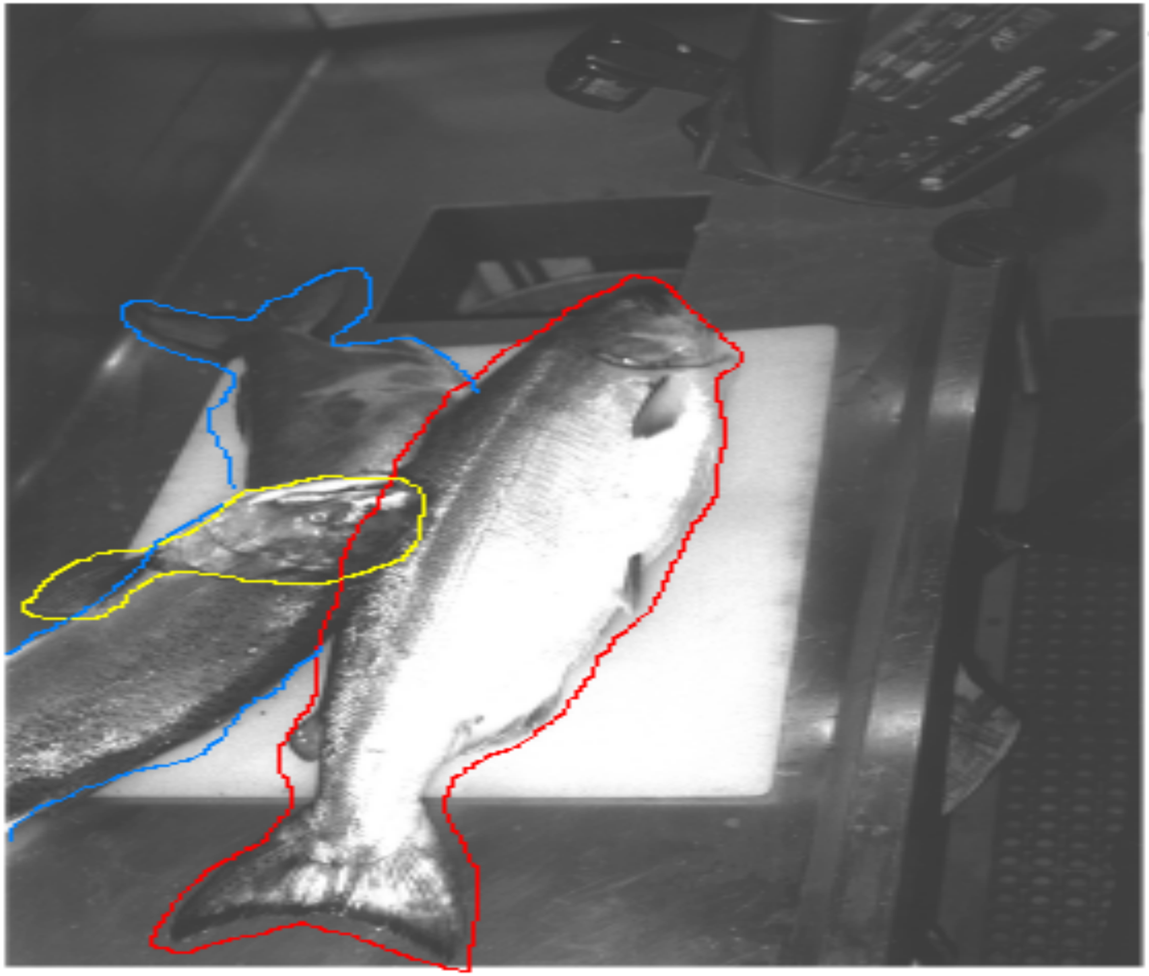
Introduction

- Statistical pattern classification is grounded into Bayesian decision theory, therefore, knowing the elements of this theory is a must for anybody wants to work in this field.
- Bayesian theory assumes that decision making problems are formulated in probabilistic terms. Probabilities are known or should be estimated.
- Bayes decision theory allows to take into account both *probability* and *"risk"* of decisions. Making a rational decision means to take into account both the probability and the risk (or the utility) associated to the decision.

- We start our presentation of elements of Bayesian decision theory assuming that all the probabilities involved in the problem considered are known.

First thing to know: the MAP decision rule

➢ We present the fundamental decision rule of Bayesian decision theory using the example of the salmon/sea bass classification introduced in Part 1.



Let us assume that an image segmentation module has already extracted the shape of the fishes as shown in the figure, and a feature extraction module has characterized each shape/pattern with one feature: the average lightness of the shape. Decision problem: we want to assign each shape/pattern to one of the two classes considered (salmon, sea bass).

First thing to know: the MAP decision rule

- We assume that we cannot know deterministically which is the "class" (salmon or sea bass) of the next fish incoming on the conveyor belt. So the problem must be formulated in probabilistic terms.
- Next incoming fish can be a salmon or a sea bass with a given probability. Bayes decision theory formalizes this situation with the concept of "state of nature" (usually called "class" in pattern recognition). In our example, we have two states-of-nature/classes: ω_1 and ω_2
- Let $\omega = \omega_1$ or $\omega = \omega_2$ be the variable that identifies the class, where ω is a *random variable*.

➢ The two classes could have the same *prior probability*:

$$P(\omega_1) = P(\omega_2)$$

$P(\omega_1) + P(\omega_2) = 1$ (we have just two species of fish)

The MAP decision rule

• If we should make a decision without being able to see the incoming fish, the only rational decision would be:

Assign the fish to ω_1 if $P(\omega_1) > P(\omega_2)$, else assign the fish to ω_2

• This "blind" (a priori) decision works well only if one class is much more likely, e.g., $P(\omega_1) >> P(\omega_2)$ (and, as we see later, the two decisions have the same risk).

➢ In general, we must "see" the pattern to make a rational decision according to Bayesian theory.

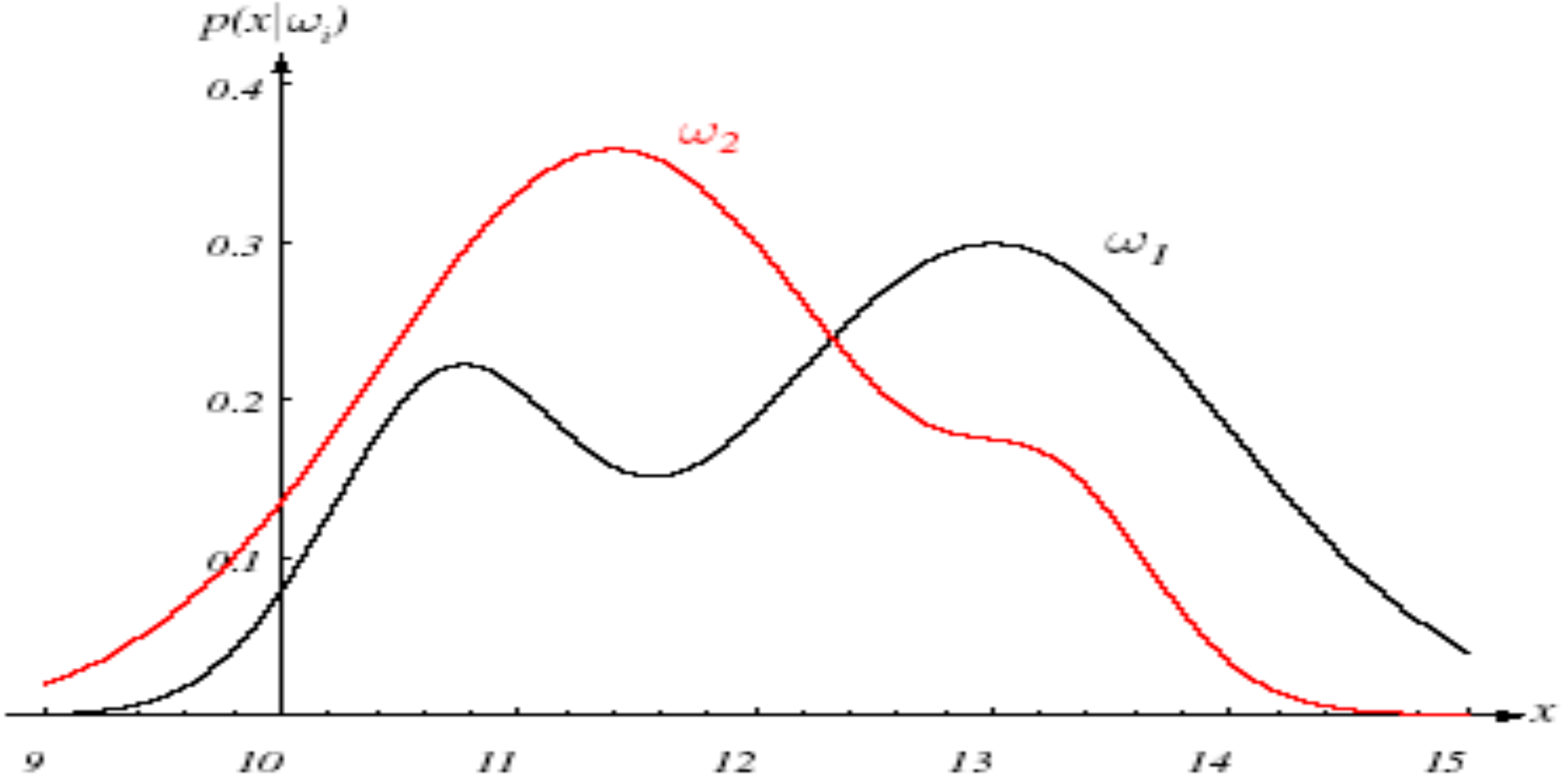
• We must see the fish and characterize it with some features.

• For example, the average lightness of the pattern.

• As fishes incoming on the belt will have "random" lightness values, the lightness feature x should be treated as a random variable with *conditional distribution* $p(x \mid \omega_i)$.

An example of a mono-dimensional $p(x \mid \omega_i)$

- $p(x \mid \omega_i)$ is the *class-conditional probability density function*



If x is the average lightness of the image region associated to a given fish of class ω_i, then the difference between the functions $p(x \mid \omega_i)$ characterizes the expected lightness difference between the two fish species.

Bayes decision rule

- Let us assume to know the two priors $P(\omega_j)$ and the two class-conditional density functions $p(x \mid \omega_j)$, $j=1,2$.
- If we measure the average lightness x of the incoming fish, taking into account the probabilistic nature of the problem, the most rationale decision rule is based on the probability:

$$P(\omega_j , x) = P(\omega_j \mid x)\, p(x) = p(x \mid \omega_j)\, P(\omega_j)$$

- That we can rewrite as the **Bayes decision rule (MAP, maximum a posteriori, decision rule)** :

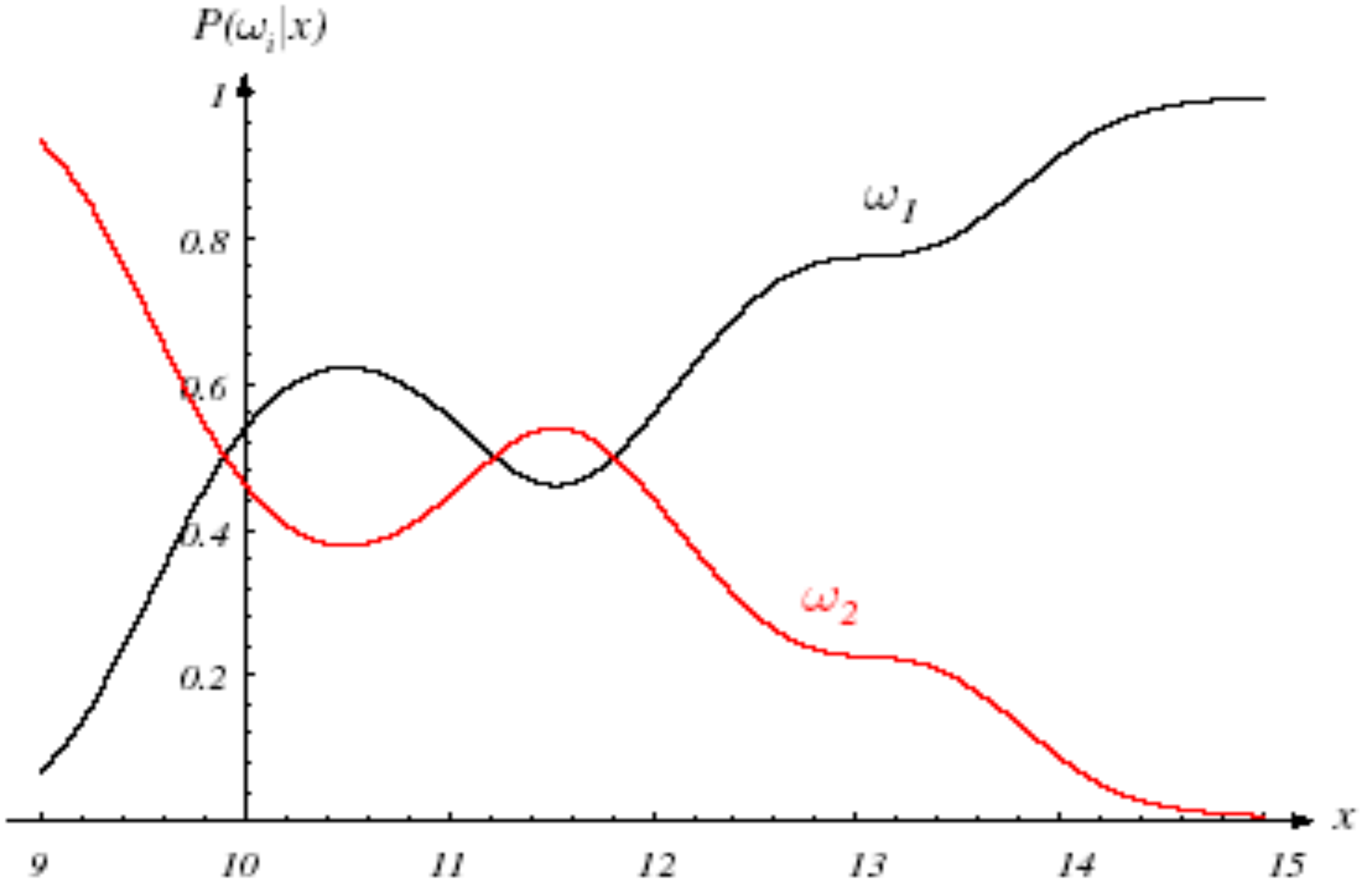
$$P(\omega_j \mid x) = p(x \mid \omega_j)\ P(\omega_j) / p(x)$$

*Posterior = (Likelihood * Prior) / Evidence*

Note that: $$p(x) = \sum_{j=1}^{2} p(x \mid \omega_j) P(\omega_j)$$

An example of mono-dimensional $P(\omega_i | x)$

$P(\omega_i | x)$ with $P(\omega_1)=2/3$ e $P(\omega_2)=1/3$



The MAP decision rule

- The MAP, maximum a posteriori probability, criterion is the most rationale decision rule for the considered probabilistic setting:

If $P(\omega_1 \mid x) > P(\omega_2 \mid x)$ then is most rationale to assign x to ω_1

If $P(\omega_1 \mid x) < P(\omega_2 \mid x)$ then is most rationale to assign x to ω_2

- This rule is the most rationale because it minimizes the error probability for any given *x*:

$$P(error \mid x) = P(\omega_1 \mid x) \text{ if we assign } x \text{ to } \omega_2$$

$$P(error \mid x) = P(\omega_2 \mid x) \text{ if we assign } x \text{ to } \omega_1$$

- We can prove that MAP rule also minimizes the average error:

$$P(errore) = \int_{-\infty}^{+\infty} P(errore, x)dx = \int_{-\infty}^{+\infty} P(errore \mid x)\, p(x)dx$$

Likelihood ratio test and ML rule

We match the likelihood ratio ***l(x)*** against a threshold θ not depending on x

- We can reformulate the MAP rule as follows:

  If $p(x / \omega_1)\, P(\omega_1) > p(x / \omega_2)\, P(\omega_2)$ then assign x to ω_1

  else assign x to ω_2

Likelihood ratio test

$$l(x) = \frac{p(x/\omega_1)}{p(x/\omega_2)} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \frac{P(\omega_2)}{P(\omega_1)} = \theta$$

- Note: the "evidence" $p(x)$ does not matter!

Note: the **likelihood ratio** is matched against the ratio between "**priors**"

Two special cases:

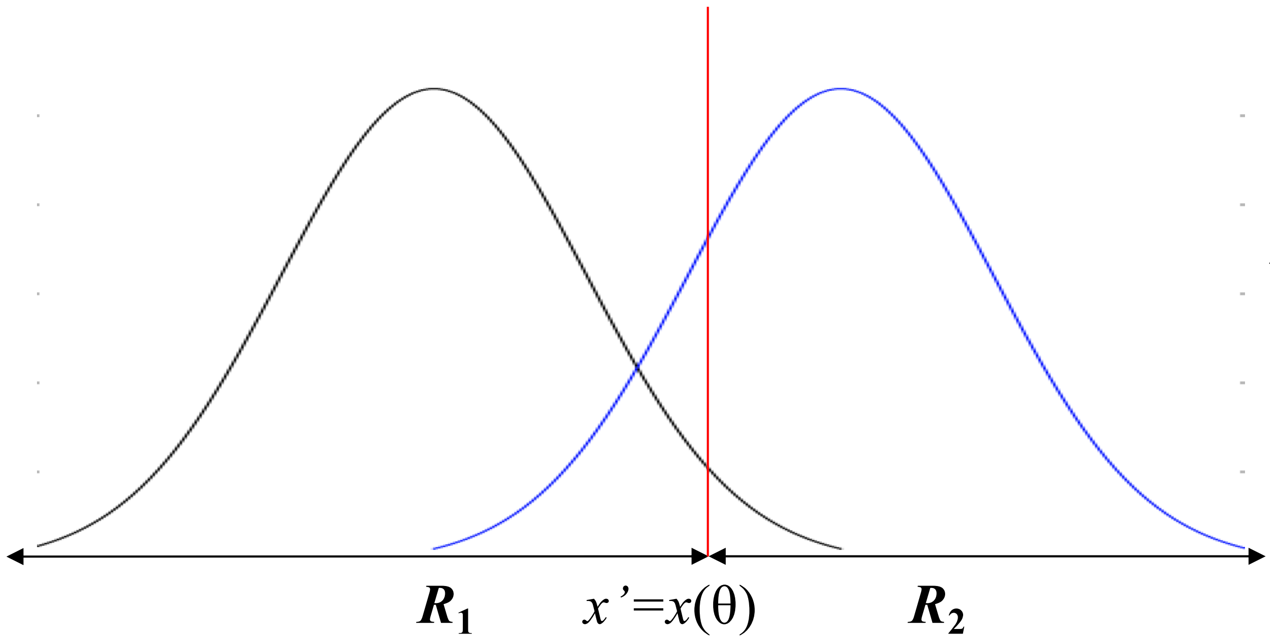
- If $p(x/\omega_1)=p(x/\omega_2)$, then the decision depends only on priors
- If $P(\omega_1)=P(\omega_2)$, then the decision depends only on likelihoods (**ML, Maximum Likelihood, decision rule**)

The concept of "decision regions"

The likelihood ratio test is defined by $l(x)$ and the threshold θ. This test identifies two decision regions R_1 e R_2 in the feature space R (here we consider a single feature x).

- $R_1 = \{\mathbf{x}\ \varepsilon\ R\colon l(\boldsymbol{x}) > \theta\}$ and $R_2 = \{\mathbf{x}\ \varepsilon\ R\colon l(\boldsymbol{x}) < \theta\}$ (if $l(\boldsymbol{x}) = \theta$ then x can be assigned randomly to R_1 or R_2).

–Given the probability density functions $p(\mathbf{x}|\omega_1)$ and $p(\mathbf{x}|\omega_2)$, the regions R_1 and R_2 are identified by the threshold θ.



A Gaussian example

$$\begin{cases} R_1 = R_1(\theta) \\ R_2 = R_2(\theta) \end{cases}$$

MAP decision rule with more than two classes

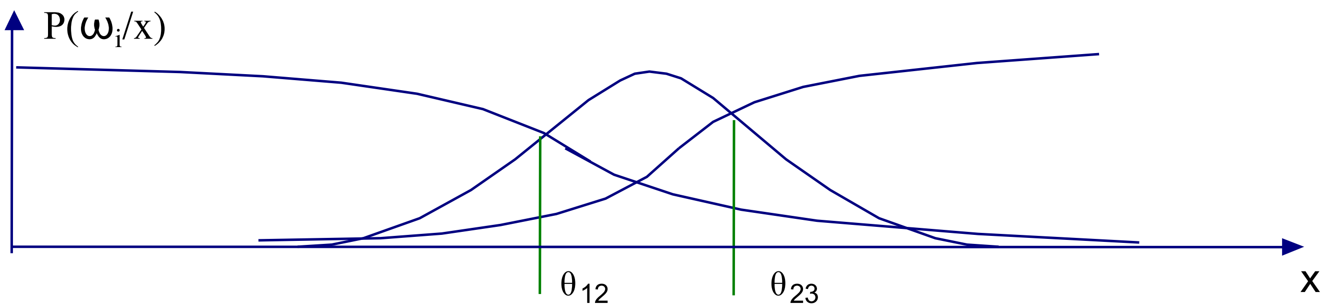
The MAP decision rule with more than two classes is:

$$\mathbf{x} \rightarrow \omega_i \Leftrightarrow P(\omega_i \mid \mathbf{x}) > P(\omega_j \mid \mathbf{x}) \ \forall \mathrm{i} \neq \mathrm{j},\ \mathrm{i}=1,\ldots,\mathrm{c}$$

- The Likelihood ratio test is defined accordingly.
- It is easy to see that we should have multiple thresholds θ_{st} defined according to the following rule :

$$P(\omega_s \mid \mathbf{x}) > P(\omega_i \mid \mathbf{x}) \ \forall s,t \neq i,\ s \neq t \ \ \mathrm{i}=1,\ldots,\mathrm{c}$$
$$P(\omega_t \mid \mathbf{x}) > P(\omega_i \mid \mathbf{x})$$

- In this example, we have three classes and two thresholds θ_{12} e θ_{23}.



Basic concepts of error probability

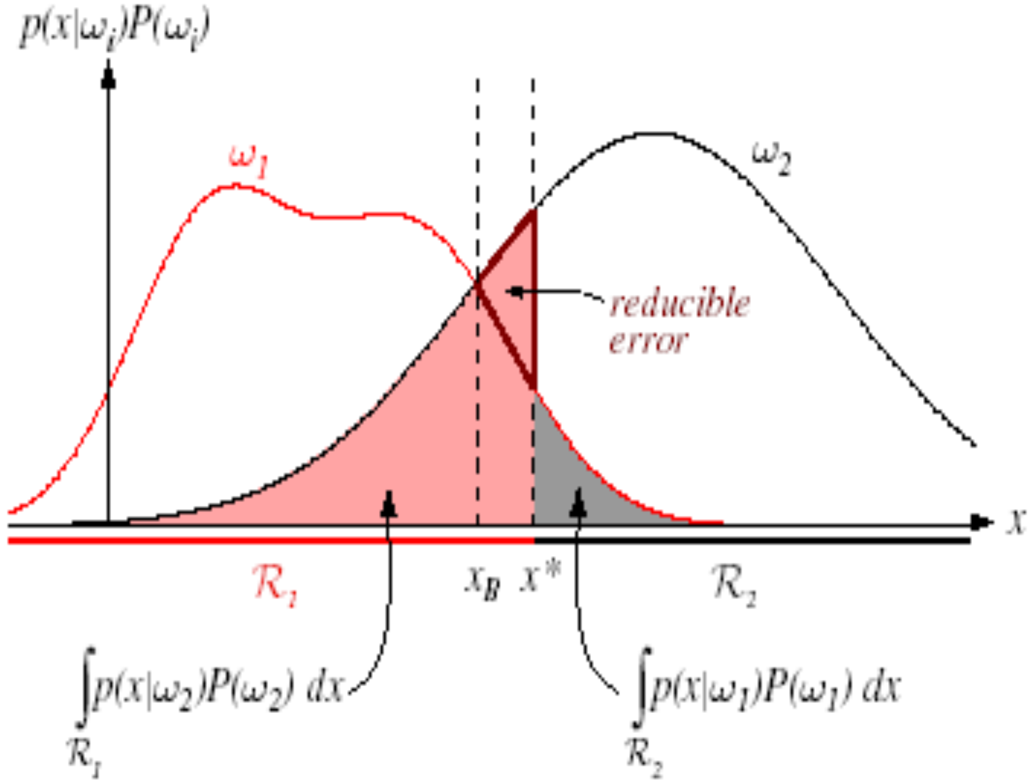
- For the two class case:

$$
\begin{aligned}
P(error) &= P\{x \in R_2, \omega_1\} + P\{x \in R_1, \omega_2\} = \\
&= P(\omega_1)P\{x \in R_2 \mid \omega_1\} + P(\omega_2)P\{x \in R_1 \mid \omega_2\} = \\
&= P(\omega_1)\int_{R_2} p(x \mid \omega_1)\,dx + P(\omega_2)\int_{R_1} p(x \mid \omega_2)\,dx
\end{aligned}
$$

The optimal threshold $x=x_B$ is the Bayesian threshold providing the minimum error, called **Bayes error**.

In the figure $x=x^*$ is a suboptimal threshold, that brings to an *added* (reducible) error over the Bayes error.

In practical cases we usually have an **added error** because the optimal threshold providing the Bayes error is nearly impossible to estimate.



Basic concepts of error probability

- With more than two classes (c>2), it is more convenient to compute the error probability by the probability of correct classification:

$$P(correct) = \sum_{i=1}^{c} P\{\mathbf{x} \in R_i, \omega_i\} = \sum_{i=1}^{c} P_i P\{\mathbf{x} \in R_i / \omega_i\} = \sum_{i=1}^{c} P_i \int_{R_i} p(\mathbf{x} \mid \omega_i) d\mathbf{x}$$

$$P(error) = 1 - P(correct)$$

In general, it is easy to see that the above computation can be very difficult, as it requires multidimensional integrals and involves density functions with very complicated analytical forms.

The computation is easy only for Gaussian densities. We will do that for exercise.

MAP rule for error probability minimization

Let us show how the MAP rule allows to minimize the error probability.

$$\mathbf{x} \rightarrow \omega_i \Leftrightarrow P(\omega_i \mid \mathbf{x}) > P(\omega_j \mid \mathbf{x}) \ \forall \mathrm{i} \neq \mathrm{j},\ \mathrm{i}=1,\ldots,\mathrm{c}$$

Error probability can be written as follows:

$$P(error) = \sum_{i=1}^{c} P(error / \omega_i) P(\omega_i)$$

With $P(error / \omega_i) = \int_{C[R_i]} p(x / \omega_i) dx$

$C\,[R_i]$ is the union of the decision regions different from R_i: $C[R_i] = \bigcup_{j=1,\, j\neq i}^{c} R_j$

MAP rule for error probability minimization

Therefore, error probability can be rewritten as:

$$P(error) = \sum_{i=1}^{c} \int_{C[R_i]} p(x/\omega_i)P(\omega_i)dx =$$

$$= \sum_{i=1}^{c} P(\omega_i)\left(1 - \int_{R_i} p(x/\omega_i)\,dx\right) = 1 - \sum_{i=1}^{c} P(\omega_i)\int_{R_i} p(x/\omega_i)\,dx$$

It is easy to see that minimization of the error probability is equal to the maximization of term related to the probability of correct classification:

$$\sum_{i=1}^{c} P(\omega_i)\int_{R_i} p(x/\omega_i)dx$$

- But this implies that decision regions R_i should be chosen in order to maximize $P(\omega_j / x) = p(x / \omega_j)P(\omega_j)$, that proves that MAP decision rule minimizes the error probability.

A quick note on error-probability upper bounds

• As exact computation of error probability is often nearly impossible, some upper bounds on the error probability have been proposed:

- **Chernoff bound**
- **Bhattacharyya bound**

✓ Students interested in further details are referred to Chapter 2.8 of the book "Pattern Classification", by R. O. Duda, P. E. Hart, and D. G. Stork, John Wiley & Sons, 2000

• However, the above bounds have been designed for Gaussian density functions. They are not reliable for non-Gaussian functions. They often are loose bounds, useful for practical applications only if the error value provided by the upper bound is acceptable (knowing that error is less than k% must be enough for your practical purposes !).

➢ We see later experimental techniques to assess error probability of a pattern classifier.

Bayesian decision theory

Now we generalize the standard setting by:

1) Allowing the use of more than one feature ("feature space"): $\mathbf{x} = (x_1, x_2, \ldots, x_d)$, feature vector with "d" elements.
2) Allowing more than two classes.
3) Introducing the concept of **"risk"**, as a generalization of the concept of error probability.
4) Allowing the **rejection** option, that is, allowing not making any decision if the decision is too risky/costly, and we can postpone it, eventually asking for human decision/intervention.

From error to risk

It is easy to see that the following expression of error probability assumes that all the errors are "equal", that is, all the terms related to probabilities of error ($j \neq i$) have the same "cost" equal to 1.

$$P(error / x \in \omega_i) = \sum_{j=1, j \neq i}^{c} P(\omega_j \mid x) = 1 - P(\omega_i \mid x)$$

For some applications, the above costs need to be different. It is easy to see that if costs are different, then the above equation is not more an error probability. The notion of risk function has been defined: $R(\omega_i / x)$

$$R(\omega_i \mid x) = \sum_{j=1}^{c} w_{ij} P(\omega_j \mid x)$$

The weights w_{ij} are the "costs" of errors. Later we denote them as $\lambda(\alpha_i / \alpha_j)$

Note that che w_{ii} can be negative ("gain")

Minimum risk theory

•MAP rule does not consider the different costs associated to the different errors

•In some applications this is not a valid choice because errors can bring different losses, and, therefore, they should have different costs.

•The minimum risk theory (also called **utility theory** in economy) takes into account both probabilities and costs of actions/decisions.

Problem formulation:

- Data classes: $\Omega = \{\omega_1, \omega_2, ..., \omega_c\}$;
- Actions/Decisions: $A = \{\alpha_1, \alpha_2, ..., \alpha_a\}$;

➢In the most of cases, we consider action=classification, that is, the action is the decision about the class of the pattern.

Minimum risk theory

The costs (losses) associated to the different actions given possible classifications are defined by the loss matrix Λ:

$$\Lambda = \begin{bmatrix} \lambda(\alpha_1 | \omega_1) & \lambda(\alpha_1 | \omega_2) & \cdots & \lambda(\alpha_1 | \omega_c) \\ \lambda(\alpha_2 | \omega_1) & \lambda(\alpha_2 | \omega_2) & \cdots & \lambda(\alpha_2 | \omega_c) \\ \vdots & \vdots & \ddots & \vdots \\ \lambda(\alpha_a | \omega_1) & \lambda(\alpha_a | \omega_1) & \cdots & \lambda(\alpha_a | \omega_c) \end{bmatrix}$$

The function $\lambda(\alpha_i|\omega_j)$ is a loss function denoting the "loss/cost" associated to the action/decision α_i when the data class is ω_j

An example of loss matrix for intrusion detection in computer networks

$\Omega=\{\omega_1=$ malicious traffic, $\omega_2=$normal traffic$\}$; $A=\{\alpha_1=$ server off, $\alpha_2=$ server on$\}$;

$$\Lambda = \begin{bmatrix} 0 & \lambda_{12} \\ \lambda_{21} & 0 \end{bmatrix}$$

Bank computer network: $\lambda_{12} << \lambda_{21}$

An example of loss matrix for intrusion detection systems

	Normal Traffic	User to Root Attack	Remote to Local Attack	Probing Attack	Denial of Service Attack
Normal Traffic	0	2	2		2
User to Root Attack	3	0	2	2	2
Remote to Local Attack	4	2	0	2	2
Probing Attack	1	2	2	0	2
Denial of Service Attack	3	2	2	1	0

Minimum risk decision rule

- Let us assume that the action α_i is candidate for execution given that the pattern **x** has been observed. We don't know the true class of the pattern **x**, but let us assume that we know $P(\omega_j|\mathbf{x})$. We can evaluate the ***conditional risk*** associated to the action α_i:

$$R(\alpha_i \mid \mathbf{x}) = \sum_{j=1}^{c} \lambda(\alpha_i \mid \omega_j) P(\omega_j \mid \mathbf{x}) = \underset{\omega \in \Omega}{E}\{\lambda(\alpha_i \mid \omega) \mid \mathbf{x}\}$$

- The conditional risk can be regarded as an average loss/cost.

Minimum risk decision rule

$$\mathbf{x} \rightarrow \alpha_i \Leftrightarrow R(\alpha_i \mid \mathbf{x}) < R(\alpha_j \mid \mathbf{x}) \ \forall \mathrm{i} \neq \mathrm{j},\ \mathrm{i}=1,\ldots,\mathrm{a}$$

Given the pattern **x**, we choose the action α_i with the minimum risk. This is the optimal decision rule for any pattern **x**.

Minimum risk for binary classification

Consider a **two class** problem and the case where **action=classification**

- Therefore, α_i correspond to assign the pattern to the class ω_i
- Let be $\lambda_{ij} = \lambda(\omega_i | \omega_j)$ the loss we incur assigning the pattern to the class ω_i when the true class is ω_j
- The conditional risk can be written as follows:

$$R(\omega_1 / \mathbf{x}) = \lambda_{11} P(\omega_1 / \mathbf{x}) + \lambda_{12} P(\omega_2 / \mathbf{x})$$
$$R(\omega_2 / \mathbf{x}) = \lambda_{21} P(\omega_1 / \mathbf{x}) + \lambda_{22} P(\omega_2 / \mathbf{x})$$

- The minimum risk decision rule is:
  - $\mathbf{x} \in \omega_1$ if $R(\omega_1 | \mathbf{x}) < R(\omega_2 | \mathbf{x})$, else $\mathbf{x} \in \omega_2$

Minimum risk for binary classification

- In terms of posterior probabilities:

$$x \in \omega_1 \text{ if } \left(\lambda_{21} - \lambda_{11}\right) P(\omega_1/\mathbf{x}) > \left(\lambda_{12} - \lambda_{22}\right) P(\omega_2/\mathbf{x})$$

- According to the Bayes rule:

$$x \in \omega_1 \text{ if } \left(\lambda_{21} - \lambda_{11}\right) p(\mathbf{x}/\omega_1) P(\omega_1) > \left(\lambda_{12} - \lambda_{22}\right) p(\mathbf{x}/\omega_2) P(\omega_2)$$

- It is reasonable to assume that $\lambda_{21} > \lambda_{11}$. If we make explicit the ratio likelihood $p(\mathbf{x}/\omega_1)/p(\mathbf{x}/\omega_2)$, the above rule can be rewritten as:

The true class is ω_1 if the likelihhod ratio is higher than a threshold θ that does not depend on **x**

$$\mathrm{x} \in \omega_1 \text{ if } l(\mathrm{x}) = \frac{p(x/\omega_1)}{p(x/\omega_2)} > \frac{\left(\lambda_{12} - \lambda_{22}\right)}{\left(\lambda_{21} - \lambda_{11}\right)} \frac{P(\omega_2)}{P(\omega_1)} = \theta$$

Minimum error and loss matrix 0-1

The action α_i corresponds to the assignment of the "pattern" $\boldsymbol{x}$ to the class ω_i.

In some cases, a simple loss function can be appropriate:

$$\lambda(\alpha_i, \omega_j) = \begin{cases} 0 & i = j \\ 1 & i \neq j \end{cases} \qquad i, j = 1, \dots, c$$

Loss matrix 0-1 or "zero-one loss function"

All the errors have the same cost equal to 1. The risk is exactly equal to the error probability:

$$R(\alpha_i \mid x) = \sum_{j=1}^{c} \lambda(\alpha_i \mid \omega_j) P(\omega_j \mid x)$$

$$= \sum_{j \neq i} P(\omega_j \mid x) = 1 - P(\omega_i \mid x)$$

Minimum error classification

Using a 0-1 loss function, the minimum risk decision rule become the classical MAP (maximum a posteriori probability):

Assign $\boldsymbol{x}$ to ω_i se $P(\omega_i \mid x) > P(\omega_j \mid x) \quad \forall j \neq i$

Rewriting in terms of the likelihood ratio:

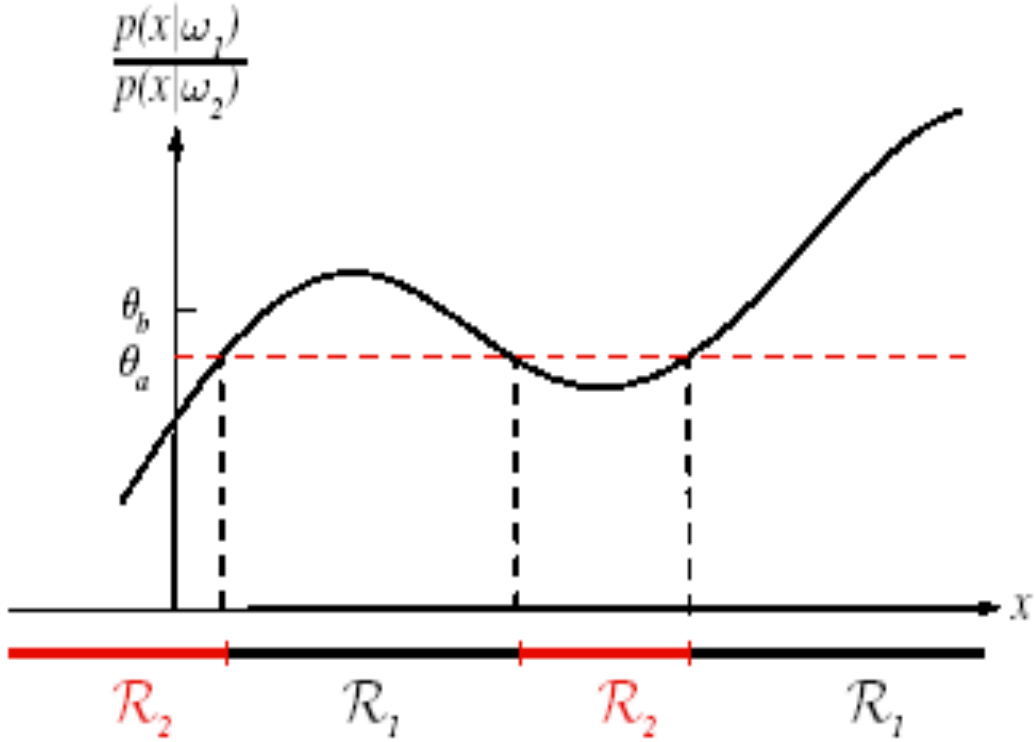
$$Given \quad \frac{\lambda_{12} - \lambda_{22}}{\lambda_{21} - \lambda_{11}} . \frac{P(\omega_2)}{P(\omega_1)} = \theta_\lambda \ ; \mathrm{x} \in \ \omega_1 \ \text{ if } \ \frac{p(x \mid \omega_1)}{p(x \mid \omega_2)} > \theta_\lambda$$

Examples

$$If\ \Lambda = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} then\ \theta_\lambda = \frac{P(\omega_2)}{P(\omega_1)} = \theta_a$$

$$If\ \Lambda = \begin{pmatrix} 0 & 2 \\ 1 & 0 \end{pmatrix} then\ \theta_\lambda = \frac{2P(\omega_2)}{P(\omega_1)} = \theta_b$$

Minimum error classification and decision regions



$$\theta = \frac{\lambda_{12}}{\lambda_{21}}$$

If errors for class ω_1 are more costly the threshold is more tight and the decision region R_1 becomes smaller

Class ω_1 should have a higher likelihood if $\lambda_{12} > \lambda_{21}$

We have the threshold θ_a when $P(\omega_1)=P(\omega_2)$ and $\lambda_{12} = \lambda_{21} = 1$

We have θ_b when $\lambda_{12} > \lambda_{21}$

The region R_1 decreases when $\lambda_{12} > \lambda_{21}$

A remark on likelihood ratio test and decision regions

Likelihood ratio test transforms the decision problem within a *d*-dimensional feature space into a mono-dimensional test against the threshold value θ, without any need to know exactly and explicitly the decision regions.

–The decision regions could be very complex manifolds, but we do not need their exact computation to classify the pattern **x**.

–To classify the pattern **x** is sufficient to compute the likelihood ration *l(**x**)* and compare it against the threshold value θ.

Some remarks on the use of minimum risk decision rule in security problems

$\Omega=\{\omega_1=$ malicious traffic, $\omega_2=$normal traffic$\}$;
A= $\{\alpha 1=$ server off, $\alpha 2=$ server on$\}$;

Loss matrix: $\Lambda = \begin{bmatrix} 0 & \lambda_{12} \\ \lambda_{21} & 0 \end{bmatrix}$

Minimum risk decision rule:

$$\text{Server off if } l(\mathrm{x}) = \frac{p(x/attack)}{p(x/normal)} > \frac{\lambda_{12}}{\lambda_{21}} \frac{P(normal)}{P(attack)} = \theta$$

We assume: $\lambda_{12} << \lambda_{21}$

- How should we set the costs?
- How should we estimate priors?

Some remarks on the use of minimum risk decision rule in security problems

- How should we set the costs?
- How should we estimate priors?

If we rewrite the decision rule as follows:

$$\theta = \frac{\lambda_{12}}{\lambda_{21}} \frac{P(normal)}{P(attack)} = \left(\frac{1}{\lambda^*}\right) P^*$$

$$l(x) = \frac{p(x / attack)}{p(x / normal)} > \left(\frac{1}{\lambda^*}\right) P^* = \theta$$

- The relationship between priors and costs/losses become much more clear
- We note that estimate P^* helps for making a decision about the costs
- If P* is large (P(normal) is much higher than P(attack) , I must set $\lambda_{12} << \lambda_{21}$ if I want that my classifier detects attacks.
- If I cannot evaluate P^* in a realiable way? We see later the Minimax rule.

Overall risk minimization

The action we do depends on the pattern **x** by the function α(**x**), and the **overall risk** *R* can be written as:

$$R = \int R\big(\alpha(\mathbf{x})/\mathbf{x}\big)p(\mathbf{x})d\mathbf{x}$$

$$R = \int \sum_{i=1}^{a} \sum_{j=1}^{c} \lambda(\alpha_i \mid \omega_j) p(\mathbf{x} / \omega_j) P(\omega_j) dx$$

•We can show that the minimum risk rule applied to any pattern **x** minimizes the overall risk over the entire feature space considered.

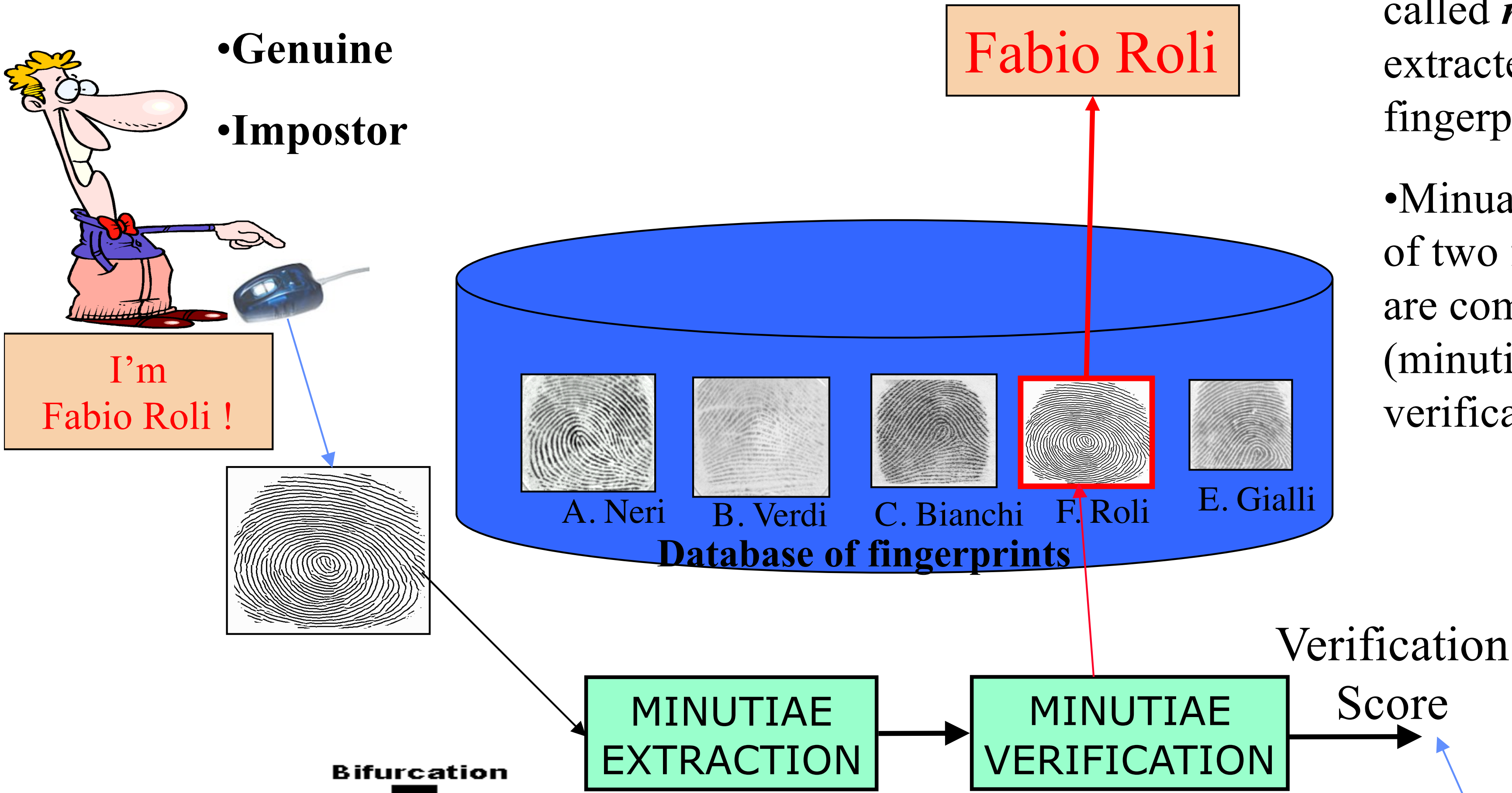
•We introduce some notable concepts (**false and miss alarm probability/rate**), and we introduce the formulation of a two class problem as a "hypothesis testing" problem

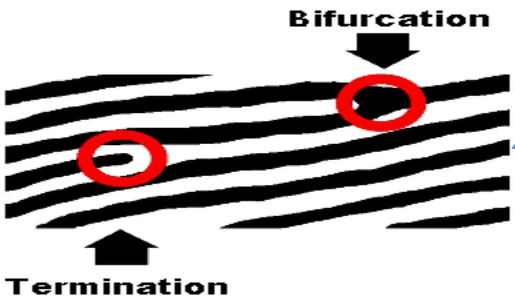
➢To analyse the overall risk minimization, let us consider the **personal identity verification** problem by **fingerprint recognition**.

Fingerprint recognition as hypothesis testing

Two class/hypothesis test:

- **Genuine**
- **Impostor**

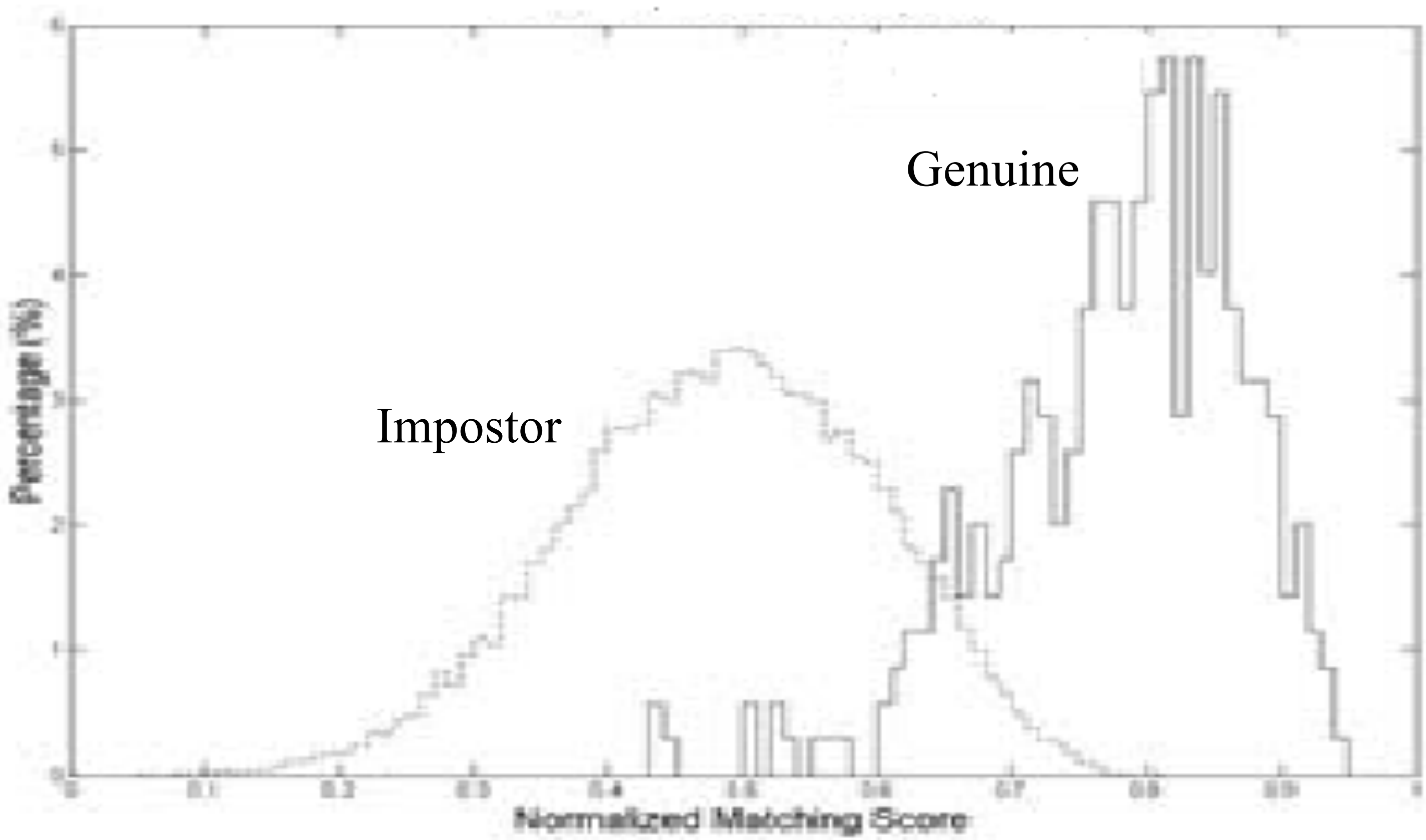




- We use *d* features, called ***minutiae***, extracted from the fingerprint image
- Minuatiae points of two fingerprints are compared (minutiae verification)

Decision is based on the "score", taking real values between 0 and 1

Decision regions



- An example of sample distributions of the "scores" of genuine and impostor users
- Decison regions R_1 e R_2 are defined according to the distributions

➢ Let us assume that p(s/genuine) and p(s/impostor) are known

Let us assume that the space *S* of the "score" values has been subdivided into the regions R_1 and R_2 so that:

If *s* belongs to R_2 then the claimed identity is verified (genuine user)

Else ($s \in R_1$) the claimed identity is rejected (impostor user)

False and miss alarm rates

The optimal decision should be associated to regions R_1 e R_2 that minimize the expected overall risk:

$$\mathcal{R} = E\{\text{risk}\} = \lambda_{11} P\left(impostor / impostor\right) P(impostor) + \lambda_{12} P\left(impostor / genuine\right) P(genuine) +$$
$$+\lambda_{21} P\left(genuine / impostor\right) P(impostor) + \lambda_{22} P\left(genuine / genuine\right) P(genuine)$$

$$P(impostor \mid impostor) = \int_{R_1} p(s \mid impostor)\, ds = 1 - P_{FA},\ P(impostor \mid genuine) = \int_{R_1} p(s \mid genuine)\, ds = P_{FA}$$

$$P(genuine \mid impostor) = \int_{R_2} p(s \mid impostor)\, ds = P_{MA},\ P(genuine \mid genuine) = \int_{R_2} p(s \mid genuine)\, ds = 1 - P_{MA}$$

$$\rightarrow\ \mathcal{R} = \lambda_{11}\left(1 - P_{FA}\right) P_{imp} + \lambda_{12} P_{FA} P_{gen} + \lambda_{21} P_{MA} P_{imp} + \lambda_{22} (1 - P_{MA}) P_{gen}$$

P_{FA}: False Allarm Rate (or False Reject Rate, FRR)

P_{MA}: Miss Allarm Rate (or False Acceptance Rate, FAR)

P_{gen}=P(genuine); P_{imp}=P(impostor);

Overall risk minimization

Rewriting the expected overall risk:

$$R = \lambda_{11} \int_{R_1} p(s \mid impostor)\, ds\ P_{imp} + \lambda_{12} \int_{R_1} p(s \mid genuine)\, ds\ P_{gen} +$$

$$+\lambda_{21} \int_{R_2} p(s \mid impostor)\, ds\ P_{imp} + \lambda_{22} \int_{R_2} p(s \mid genuine)\, ds\ P_{gen}$$

$$\sin ce \int_{R_2} p(s \mid impostor)\, ds = 1 - \int_{R_1} p(s \mid impostor)\, ds \quad \int_{R_2} p(s \mid genuine)\, ds = 1 - \int_{R_1} p(s \mid genuine)\, ds$$

$$R = \lambda_{11} \int_{R_1} p(s \mid impostor)\, ds\ P_{imp} + \lambda_{12} \int_{R_1} p(s \mid genuine)\, ds\ P_{gen} +$$

$$+\lambda_{21} P_{imp} - \lambda_{21} P_{imp} \int_{R_1} p(s \mid impostor)\, ds + \lambda_{22} P_{gen} - \lambda_{22} P_{gen} \int_{R_1} p(s \mid genuine)\, ds =$$

$$\lambda_{21} P_{imp} + \lambda_{22} P_{gen} + \int_{R_1} P_{imp}(\lambda_{11} - \lambda_{21}) p(s \mid impostor) + P_{gen}(\lambda_{12} - \lambda_{22}) p(s \mid genuine)\, ds$$

- The integrand should be negative in order to minimize the risk !

Overall risk minimization

$$P_{imp}(\lambda_{11}-\lambda_{21})p(s\,|\,impostor)+P_{gen}(\lambda_{12}-\lambda_{22})p(s\,|\,genuine)<0$$

$$\rightarrow P_{imp}(\lambda_{21}-\lambda_{11})p(s\,|\,impostor)>P_{gen}(\lambda_{12}-\lambda_{22})p(s\,|\,genuine)$$

$$\rightarrow \frac{p(s\,|\,impostor)}{p(s\,|\,genuine)}>\frac{P_{gen}}{P_{imp}}\frac{(\lambda_{12}-\lambda_{22})}{(\lambda_{21}-\lambda_{11})}$$

- The above is the minimum risk decison rule to be used for any "pattern" "s"
- This proves that that the minimum risk decison rule, used for any "pattern" "s", minimizes the expected overall risk.

Minimax decision rule

- In some real cases, priors can change over time, or their estimation can be difficult (intrusion detection, spamming, ecc.). We need a decision rule that can work even if the priors are not known
- An approach (used in many engineering problems) is based on the worst-case design
- We design the classifier in order to minimize the risk in the worst-case in terms of priors variation
- **Minimax**: **Mimize** the **Maximum** risk
- This is worst-case design. The design is very conservative/pessimistic, and therefore, performance are not the optimal ones, they are optimal only for the worst case.

Minimax decision rule

Let us consider a two class problem and two regions (not known at the beginning) R_1 and R_2. The overall risk can be written as:

$$R = \int_{\Re_1} \left(\lambda_{11} \cdot P_1 \cdot p(x/\omega_1) + \lambda_{12} \cdot P_2 \cdot p(x/\omega_2) \right) dx + \qquad \begin{cases} P_1 = P(\omega_1) \\ P_2 = P(\omega_2) \end{cases}$$

$$+ \int_{\Re_2} \left(\lambda_{21} \cdot P_1 \cdot p(x/\omega_1) + \lambda_{22} \cdot P_2 \cdot p(x/\omega_2) \right) dx$$

We know that $P_2 = 1 - P_1$ and $\int_{\Re_1} p(x/\omega_1) dx = 1 - \int_{\Re_2} p(x/\omega_1) dx$

Note: we express R as a function of P_1 and simplify the equation

$$R(P_1) = \lambda_{22} + (\lambda_{12} - \lambda_{22}) \int_{\Re_1} p(x/\omega_2)\, dx +$$

$$+ P_1 \left[(\lambda_{11} - \lambda_{22}) + (\lambda_{21} - \lambda_{11}) \int_{\Re_2} p(x/\omega_1) dx + (\lambda_{22} - \lambda_{12}) \int_{\Re_1} p(x/\omega_2) dx \right]$$

Minimax

$$R(P_1) = \lambda_{22} + (\lambda_{12} - \lambda_{22}) \int_{\Re_1} p(x/\omega_2)\,dx + P_1 \cdot \left[\ldots + \int_{\Re_2} \ldots + \int_{\Re_1} \ldots \right]$$

Note that costs and priors identifies the threshold θ. The threshold and the density functions defines the regions R_1 e R_2 and the overall risk.

$$l(x) = \frac{p(x/\omega_1)}{p(x/\omega_2)} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \frac{(\lambda_{12} - \lambda_{22})}{(\lambda_{21} - \lambda_{11})} \frac{P(\omega_2)}{P(\omega_1)} = \theta$$

$$R_1 = \{x : l(x) > \theta\}, R_2 = \{x : l(x) < \theta\}$$

- Varying P_1 changes the threshold $\theta(P_1)$, the decision regions, and the overall risk. This does not allow to control the risk! This is the key point!

➢ Key issue: P_1 can change! I want to estimate the risk I could incur, that is, I want to evaluate it not depending on P_1 variations.

Minimax

$$R(P_1) = \underbrace{\lambda_{22} + (\lambda_{12} - \lambda_{22}) \int_{\Re_1} p(x/\omega_2)\, dx}_{R_{mm}\text{, minimax risk}} + P_1 \cdot \left[\ldots + \int_{\Re_2} \ldots + \int_{\Re_1} \ldots \right]$$

The above equation shows that after identifying the decision regions (R_1 and R_2), overall risk is a linear function of P_1.

If R_1 e R_2 makes zero the term in square brackets, then the overall risk does not depend on priors! Key point!

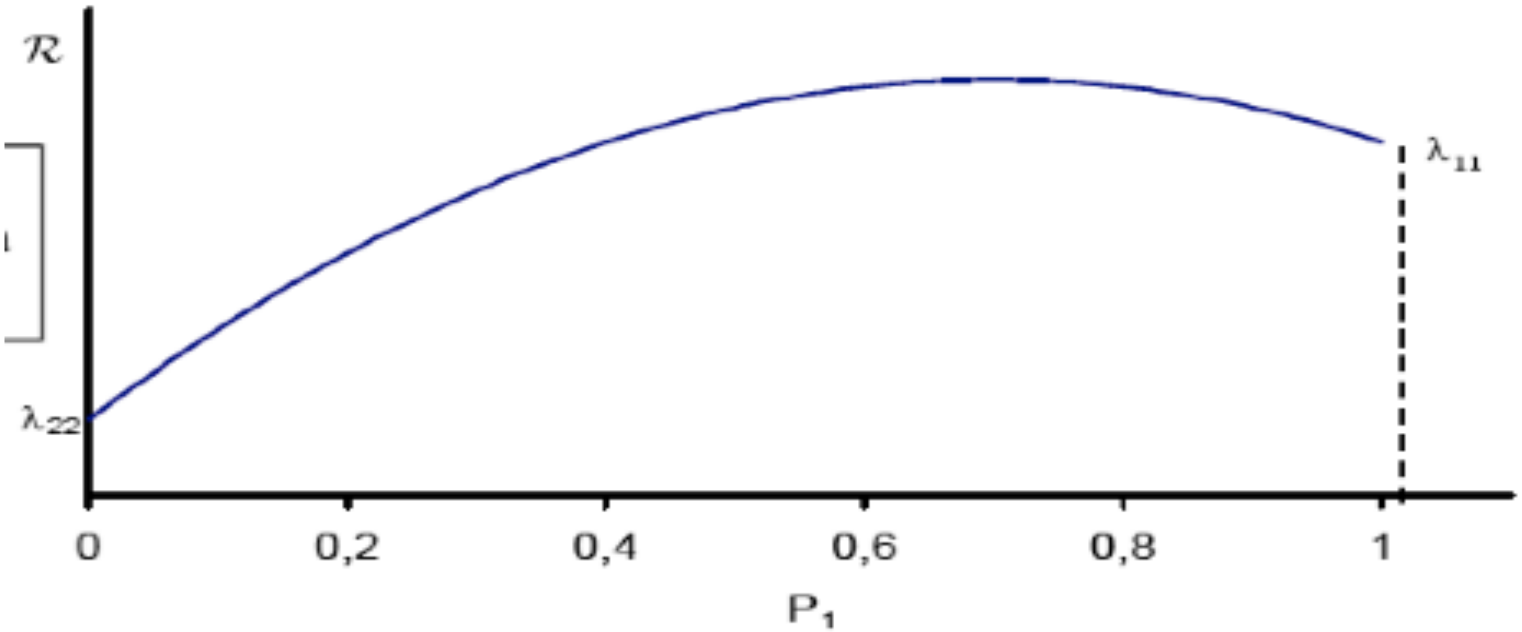
This is the **minimax** solution, and the minimax risk, R_{mm}, is:

$$R_{mm} = \lambda_{22} + (\lambda_{12} - \lambda_{22}) \int_{\Re_1} p(x/\omega_2)\, dx =$$

$$= \lambda_{11} + (\lambda_{21} - \lambda_{11}) \int_{\Re_2} p(x/\omega_1)\, dx$$

Check this equality by exercise!

Minimax: risk as a function of P_1

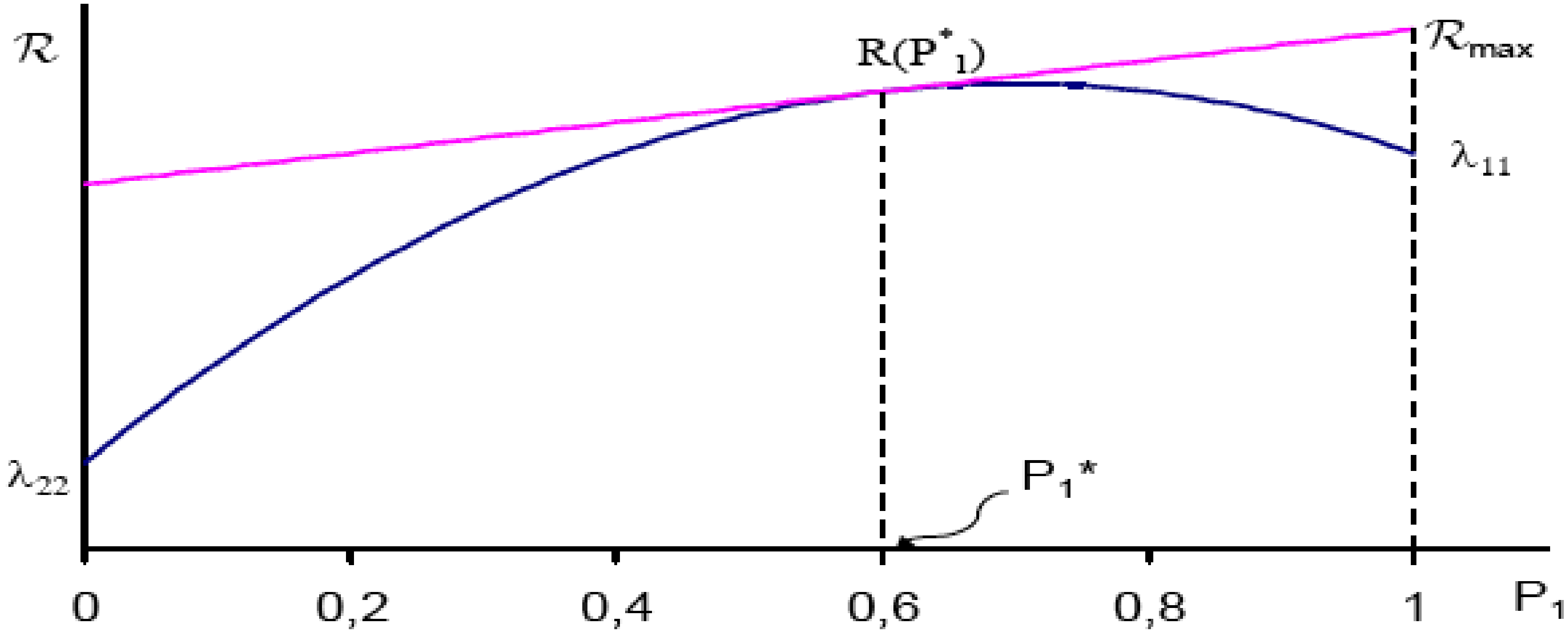


From the formula of $R(P_1)$ it is easy to see that:

$P_1 = 0$ implies that region R_1 is empty, therefore $R=\lambda_{22}$

$P_1 = 1$ implies that region R_2 is empty, therefore $R=\lambda_{11}$

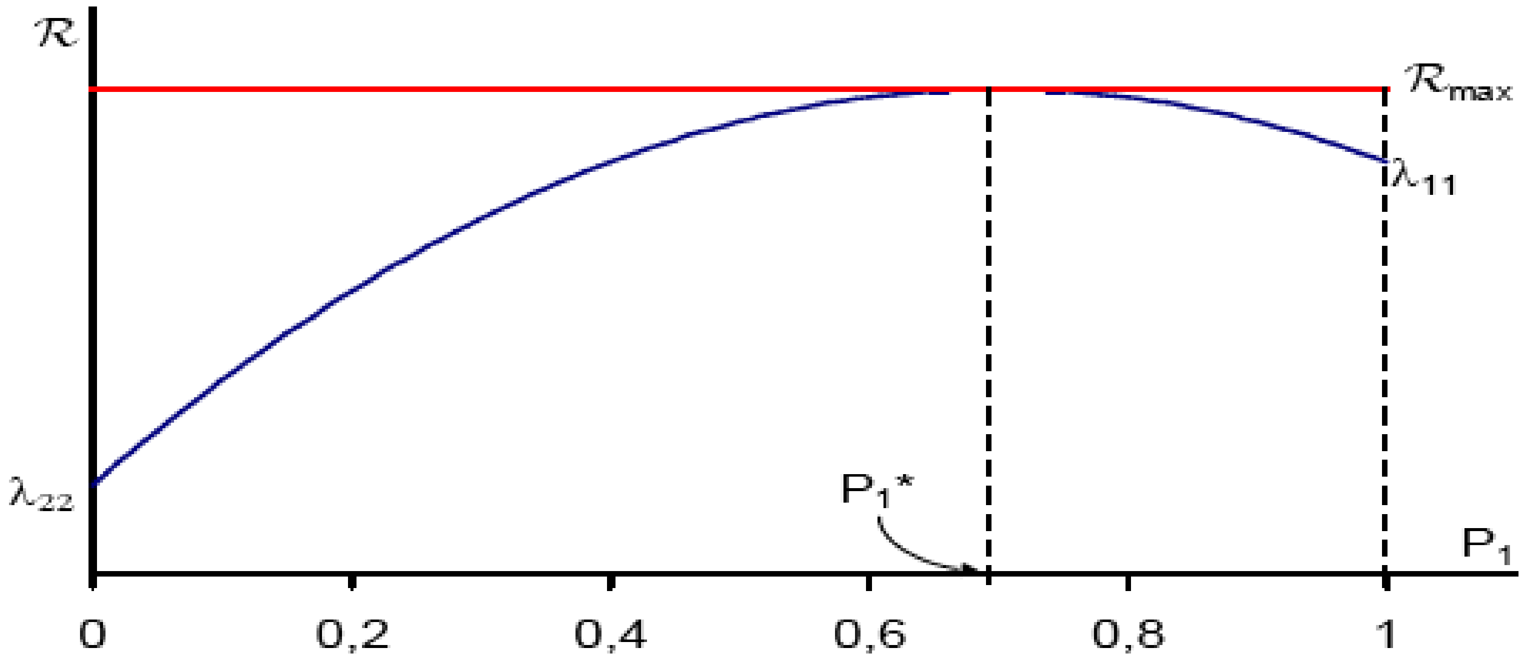
Minimax: risk linear function



Let us assume that we have $P(\omega_1)=P_1^*=0.6$, and therefore the risk associated to decision regions $R_1(P_1^*)$ and $R_2(P_1^*)$ is $R(P_1^*)$

If P_1 changes over time, the above equation shows that risk is a linear function, with decision regions identified by $P_1^*=0.6$.

Minimax: linear risk function

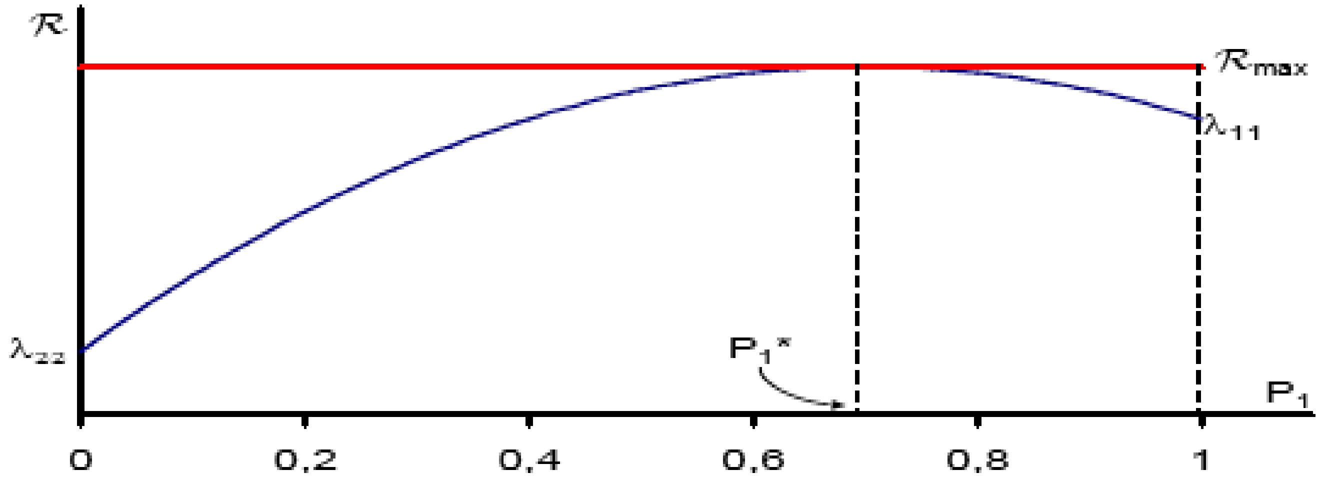


To control the risk we choose P_1* in order to have the red linear function with slope=0

The related regions $R_1(P_1^*)$ e $R_2(P_1^*)$ provides R_{max}

We are minimizing the maximum risk that we can incur when priors changes (Minimax). Indeed any other rule can provide higher risk when the prior P_1* changes (any other rule is a linear function that can provide higher risk for P_1* increasing).

Minimax



In order to identify the regions $R_1(\theta)$ e $R_2(\theta)$ associated to the Minimax line:

$$\left[(\lambda_{11}-\lambda_{22})+(\lambda_{21}-\lambda_{11})\int_{\Re_2} p(x/\omega_1)dx+(\lambda_{22}-\lambda_{12})\int_{\Re_1} p(x/\omega_2)dx\right]=0$$

- We should identify the regions that meet the above equation.
- This can be done in closed forms in a few cases, but it is very difficult in the most of real cases (see the next slide for the case of costs 0-1).

Minimax line with costs 0-1

In order to identify the regions $R_1(\theta)$ e $R_2(\theta)$ associated to the Minimax line:

$$\left[(\lambda_{11} - \lambda_{22}) + (\lambda_{21} - \lambda_{11}) \int_{\Re_2} p(x/\omega_1)dx + (\lambda_{22} - \lambda_{12}) \int_{\Re_1} p(x/\omega_2)dx \right] = 0$$

- For loss matrix 0-1, we have:

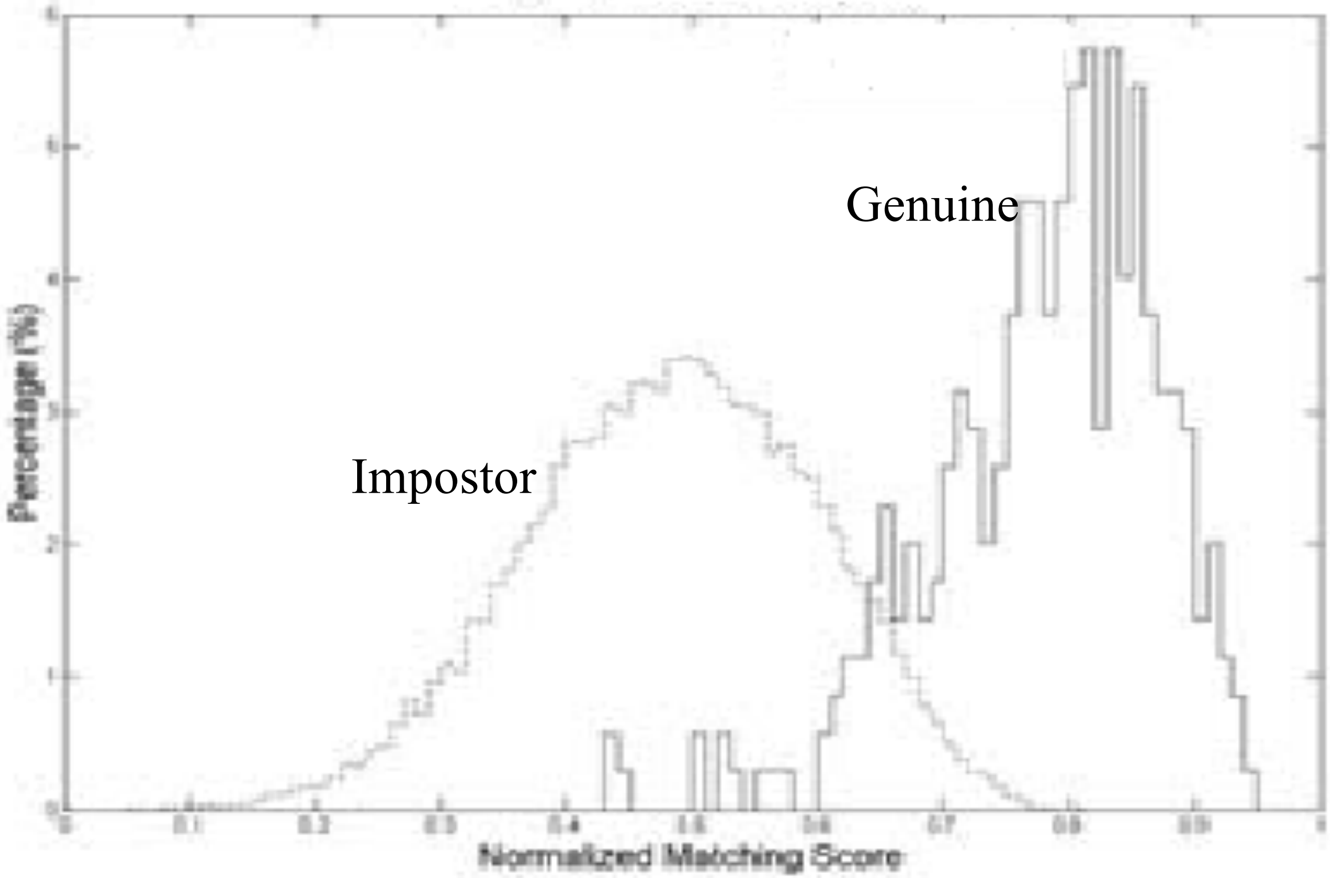
$$\int_{\Re_2} p(x/\omega_1)dx - \int_{\Re_1} p(x/\omega_2)dx = 0 \rightarrow \int_{\Re_2} p(x/\omega_1)dx = \int_{\Re_1} p(x/\omega_2)dx$$

Minimax line with costs 0-1

- For loss matrix 0-1

$$\int_{\Re_2} p(x/\omega_1)dx - \int_{\Re_1} p(x/\omega_2)dx = 0 \rightarrow \int_{\Re_2} p(x/\omega_1)dx = \int_{\Re_1} p(x/\omega_2)dx$$

The mini-max threshold $\theta*$ is the one that makes equal the two error probabilities ($P_{FA}=P_{MA}$). In fingerprint recognition it is called EER (Equal Error Rate) threshold.



Note that the threshold $\theta*$ that makes $P_{FA}=P_{MA}$ is not the optimal Bayesian threshold. Indeed we are using the Minimax rule, that is not the ideal minimum risk rule.

Minimax straight line for the general case

$$\left[(\lambda_{11}-\lambda_{22}) + (\lambda_{21}-\lambda_{11}) \int_{\Re_2} p(x/\omega_1)dx + (\lambda_{22}-\lambda_{12}) \int_{\Re_1} p(x/\omega_2)dx \right] = 0$$

➢The above integrals are the error probabilities associated to the two classess (they are the so called P_{FA} e P_{MA}), it is easy to see that these errors can be controlled by θ. We could look for a threshold θ∗ which meet the above equation. This threshold value identifies $R_1(\theta*)$ and $R_2(\theta*)$.

$$R_1 = \{x : l(x) > \theta *\}, R_2 = \{x : l(x) < \theta *\}$$

Note that:

$$\theta* \neq \frac{(\lambda_{12}-\lambda_{22})}{(\lambda_{21}-\lambda_{11})} \frac{P(\omega_2)}{P(\omega_1)}$$

We are looking for the optimal threshold θ∗ without using the priors and the costs!

Minimax line: general case

$$\left[(\lambda_{11}-\lambda_{22})+(\lambda_{21}-\lambda_{11})\int_{\Re_2} p(x/\omega_1)dx+(\lambda_{22}-\lambda_{12})\int_{\Re_1} p(x/\omega_2)dx\right]=0$$

- In general, the empirical computation of the minimax line demands for a classifier that allows the “control” of the regions R_1 e R_2.
- This is not always doable
- If the loss matrix has values 0-1 we can change the “parameters” of the classifier so that:

$$\int_{\Re_2} p(x/\omega_1)dx = \int_{\Re_1} p(x/\omega_2)dx$$

- That is, we tune the parameters in order to make equal the two error probabilities

Neyman-Pearson decision rule

- If we do not know priors and costs, we can use the Neyman-Pearson decision rule.
- This rule is used for applications where we have a constraint on the false alarm rate and we want to minimize the miss alarm rate (e.g., in radar applications or biometric recognition)
  - We fix a given P_{FA} (false alarm rate), $P_{FA} = \alpha$.
  - The Neyman-Pearson rule minimizes P_{MA} with $P_{FA} = \alpha$.
  - The minimization problem is formulated as follows:

$$F = P_{MA} + \lambda(P_{FA} - \alpha) = \int_{R_2} p(\mathbf{x} | \omega_1) d\mathbf{x} + \lambda\left[\int_{R_1} p(\mathbf{x} | \omega_2) d\mathbf{x} - \alpha\right] =$$

$$= \int_{R_2} p(\mathbf{x} | \omega_1) d\mathbf{x} + \lambda\left[1 - \int_{R_2} p(\mathbf{x} | \omega_2) d\mathbf{x} - \alpha\right] =$$

$$= \lambda(1-\alpha) + \int_{R_2} [p(\mathbf{x} | \omega_1) - \lambda p(\mathbf{x} | \omega_2)] d\mathbf{x}$$

Neyman-Pearson rule for a two class problem

Problem: identify the region R_2 that solves the above constrained minimization.

- We can disregard the constant term, and the minimization problem becomes:

$$\begin{cases} \min\limits_{R_2 \subset R} \int\limits_{R_2} [p(\mathbf{x} \mid \omega_1) - \lambda p(\mathbf{x} \mid \omega_2)] d\mathbf{x} \\ P_{FA} = \int\limits_{R_1} p(\mathbf{x} \mid \omega_2) d\mathbf{x} = \alpha \end{cases}$$

- The terms in the squared brackets are positive: so we have the minimum when the integrand is negative for any $\mathbf{x} \, \varepsilon \, R_2$. So $R_2 = \{\mathbf{x} \, \varepsilon \, R: p(\mathbf{x} | \omega_1) < \lambda \, p(\mathbf{x} | \omega_2)\} = \{\mathbf{x} \, \varepsilon \, R: l(x) < \lambda\}$.
- Therefore the decision rule is:

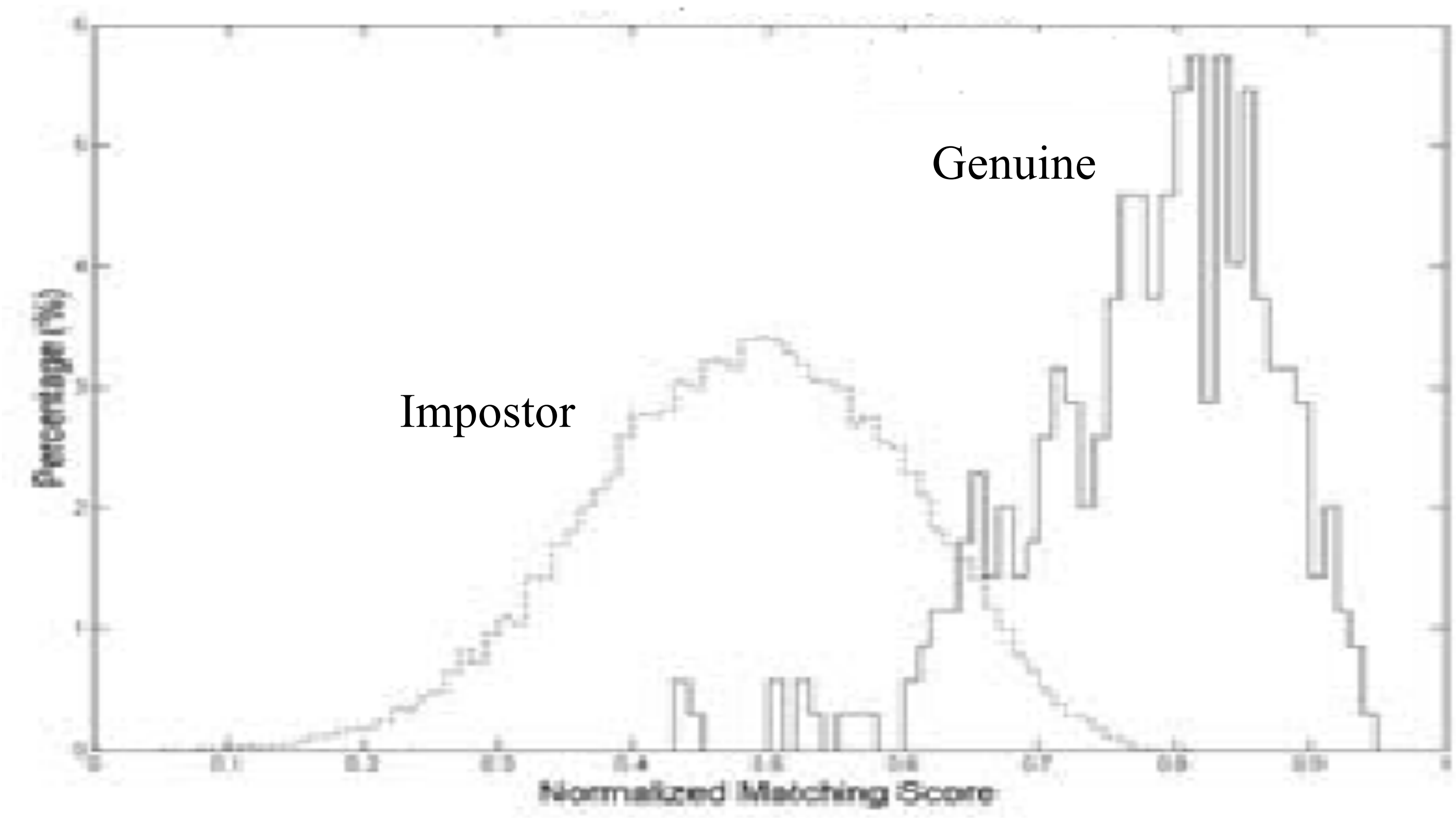
$$l(\mathbf{x}) = \frac{p(\mathbf{x} \mid \omega_1)}{p(\mathbf{x} \mid \omega_2)} \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \lambda$$

Neyman-Pearson rule: threshold computation

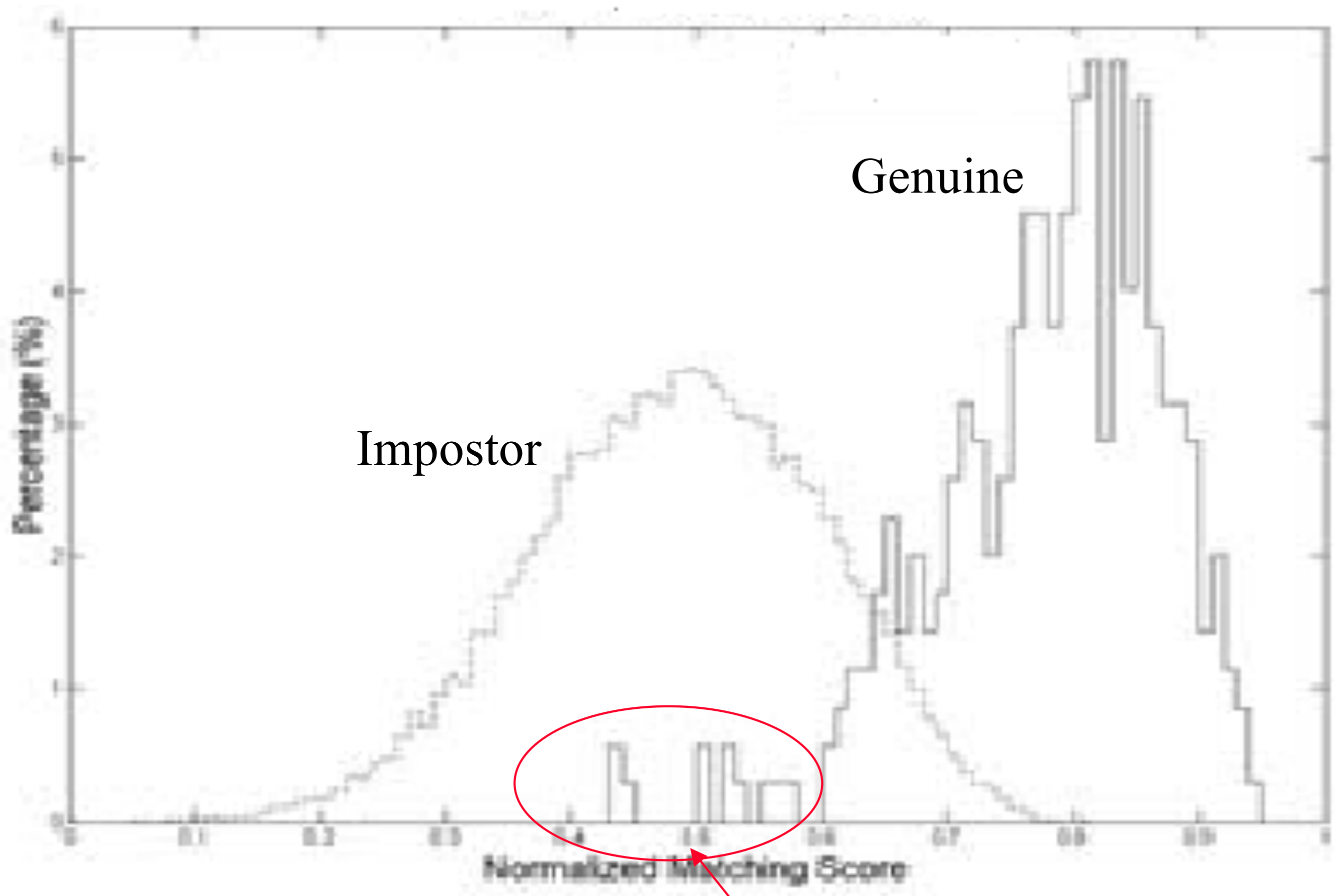
The Neyman-Pearson is based on a likelihood ratio test. The threshold of this test comes from the constraint you have.

- As $P_{FA} = P_{FA}(\lambda)$, the constraint $P_{FA} = \lambda$ identifies the values of λ.

➢In practice, the following costraint is met by experiments: $P_{FA} \leq \alpha$



Neyman-Pearson: biometric example



In this case it is easy finding the threshold that provides $P_{FA} \leq \alpha$

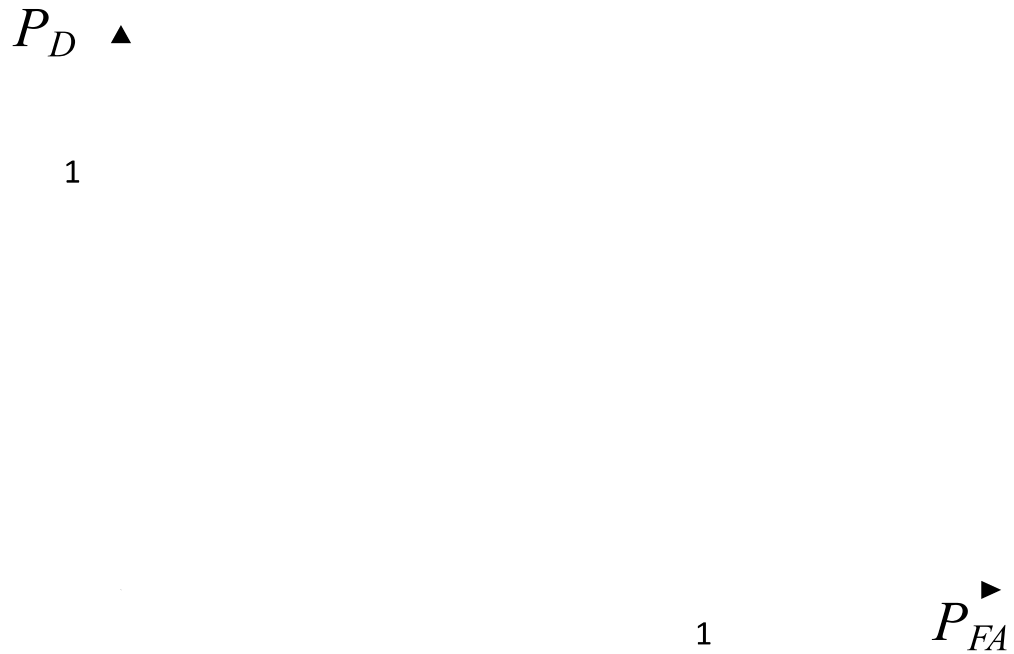
The threshold value is identified by experiments

Neyman-Pearson rule and ROC curve

With Neyman-Pearson rule, varying the threshold λ we obtain different P_{FA} and different values of probability of detection P_D (P_D=1- P_{MA}).

To assess performance for varying threshold values one can use the ROC (**Receiver Operating Characteristic**) curve.

The ROC curve shows P_D as a function of P_{FA} for different threshold.



ROC examples

✓The ROC curve allows to analyse the trade-off between the false and miss alarm rates.

✓A good ROC curve should have small P_{FA} and large P_D value.

✓The ideal ROC curve is a "rectangular" curve.

Decision with reject option

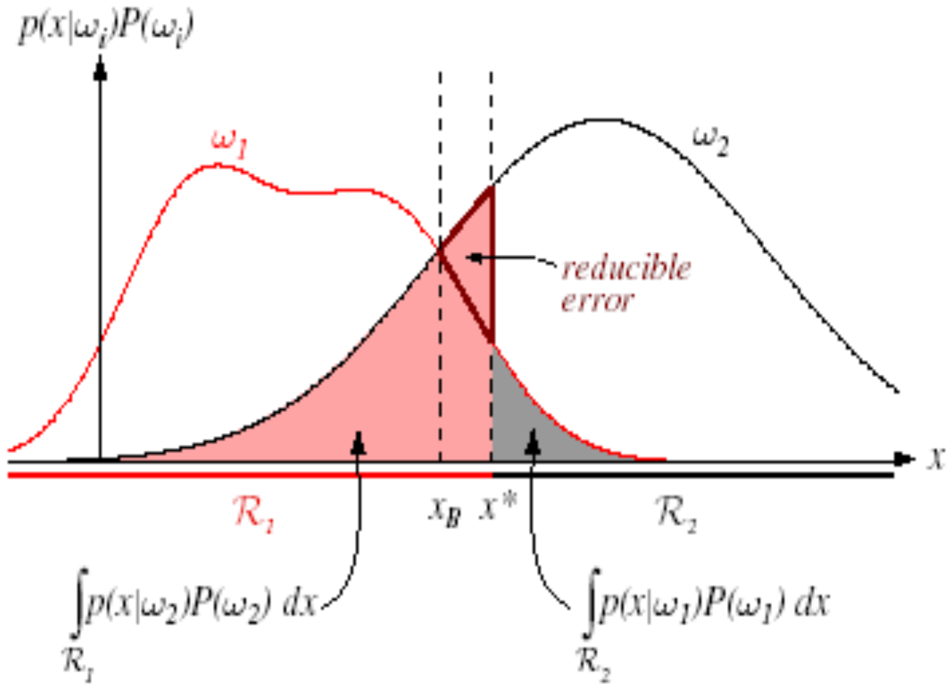
•Even if one would be able to obtain the minimum Bayes error (threshold x_B in figure), this error rate could be not acceptable for a given application

•Example: "screening" for medical diagnosis. I could demand for a "false negative" rate equal to zero.

•It is easy to see that this happens if

✓Errors are very costly, so we must protect against errors, limiting the error rate below a given threshold

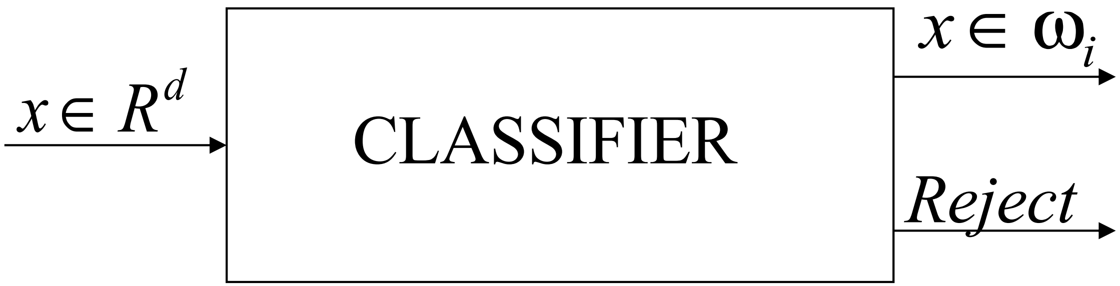
✓An obvious way to limit decision errors is not making a decision or postponing decisions



➢To reduce error probability one can omit or defer decisions (**reject option**)

➢Omitting decisions is a rationale and doable option supposed that decisions are taken by other ways (e.g., by humans)

Classification with reject option



It is easy to see that rejection option demands for an additional class with respect to the standard formulation of the classification problem:

– Set of classes: $\Omega = \{\omega_1, \omega_2, ..., \omega_c\}$;
– Set of actions/decisions: $A = \{\alpha_o, \alpha_1, \alpha_2, ..., \alpha_a\}$;
– If our action is a classification: $A = \{\omega_o, \omega_1, \omega_2, ..., \omega_c\}$;

➢ We have an additional class: ω_o, the class containing the rejected samples

Loss matrix and minimum risk with reject option

The loss matrix Λ, with size *(c+1)x c*, is:

$$\Lambda = \begin{bmatrix} \lambda(\omega_0 | \omega_1) & \lambda(\omega_0 | \omega_2) & \cdots & \lambda(\omega_0 | \omega_c) \\ \lambda(\omega_1 | \omega_1) & \lambda(\omega_1 | \omega_2) & \cdots & \lambda(\omega_1 | \omega_c) \\ \vdots & \vdots & \ddots & \vdots \\ \lambda(\omega_c | \omega_1) & \lambda(\omega_c | \omega_1) & \cdots & \lambda(\omega_c | \omega_c) \end{bmatrix}$$

The minimum risk decision criterion is:

$$\mathbf{x} \rightarrow \omega_i \Leftrightarrow R(\omega_i \mid \mathbf{x}) < R(\omega_j \mid \mathbf{x}) \ \forall \mathrm{i} \neq \mathrm{j},\ \mathrm{i}=0,1,..,\mathrm{c}$$

The main difference w.r.t. the case without reject option is that the minimum risk decision could be a “rejection”, if:

$$R(\omega_0 \mid \mathbf{x}) < R(\omega_j \mid \mathbf{x}) \ \forall j \neq 0$$

Binary classification with equal costs

Let us consider a binary classification with equal costs:

$$\Lambda = \begin{pmatrix} \lambda_r & \lambda_r \\ \lambda_c & \lambda_e \\ \\ \lambda_c & \lambda_e \end{pmatrix} = \begin{pmatrix} \lambda(\omega_0 \mid \omega_1) & \lambda(\omega_0 \mid \omega_2) \\ \lambda(\omega_1 \mid \omega_1) & \lambda(\omega_1 \mid \omega_2) \\ \\ \lambda(\omega_2 \mid \omega_2) & \lambda(\omega_2 \mid \omega_1) \end{pmatrix}$$

Reject cost = λ_r Error cost = λ_e Cost of correct classification = λ_c (usually λ_c=0)

According to the minumum risk criterion, we have three decision regions:

$$R_0 = \left\{ x \in R : R(\omega_0 \mid \mathbf{x}) < R(\omega_j \mid \mathbf{x}) \ \forall j \neq 0 \right\}$$

$$R_1 = \left\{ x \in R : R(\omega_1 \mid \mathbf{x}) < R(\omega_j \mid \mathbf{x}) \ \forall j \neq 1 \right\}$$

$$R_2 = \left\{ x \in R : R(\omega_2 \mid \mathbf{x}) < R(\omega_j \mid \mathbf{x}) \ \forall j \neq 2 \right\}$$

Binary classification with equal costs

The overall risk R can be written as:

$$R = \lambda_r \left[P(rigetto, x \in \omega_1) + P(rigetto, x \in \omega_2)\right] +$$

$$+\lambda_e \left[P(errore, x \in \omega_1) + P(errore, x \in \omega_2)\right] +$$

$$+\lambda_c \left[P(corretto, x \in \omega_1) + P(corretto, x \in \omega_2)\right] =$$

$$= \lambda_r \left[P(\omega_1)\int_{R_0} p(x/\omega_1)dx + P(\omega_2)\int_{R_0} p(x/\omega_2)dx + \right] +$$

$$+\lambda_e \left[P(\omega_1)\int_{R_2} p(x/\omega_1)dx + P(\omega_2)\int_{R_1} p(x/\omega_2)dx + \right] +$$

$$+\lambda_c \left[P(\omega_1)\int_{R_1} p(x/\omega_1)dx + P(\omega_2)\int_{R_2} p(x/\omega_2)dx + \right]$$

The error-reject trade-off

From the previous equation, we obtain:

$$R = \lambda_r P(reject) + \lambda_e P(error) + \lambda_c P(correct)$$

$$being \; P(reject) + P(error) + P(correct) = 1 \; we \; have$$

$$R = (\lambda_r - \lambda_c) P(reject) + (\lambda_e - \lambda_c) P(error)$$

- The above formulation of the overall risk clearly shows that a given value of the risk can be obtained by a trade-off between error probability and reject probability: **error-reject trade-off**

- The trade-off is also clearly shown by the relation of the three error probabilities:

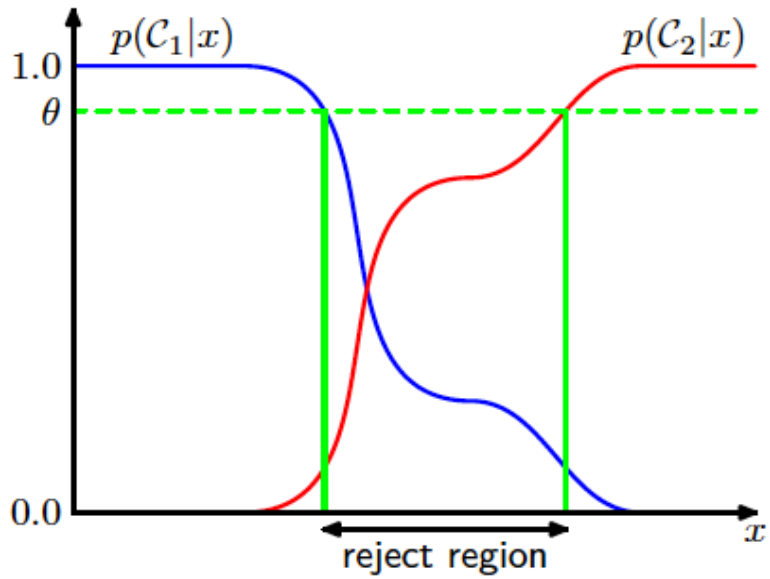
P(error)=1-P(reject)-P(correct)

- The above equation shows that we can reduce error probability by increasing reject probability.

Illustration of the reject option

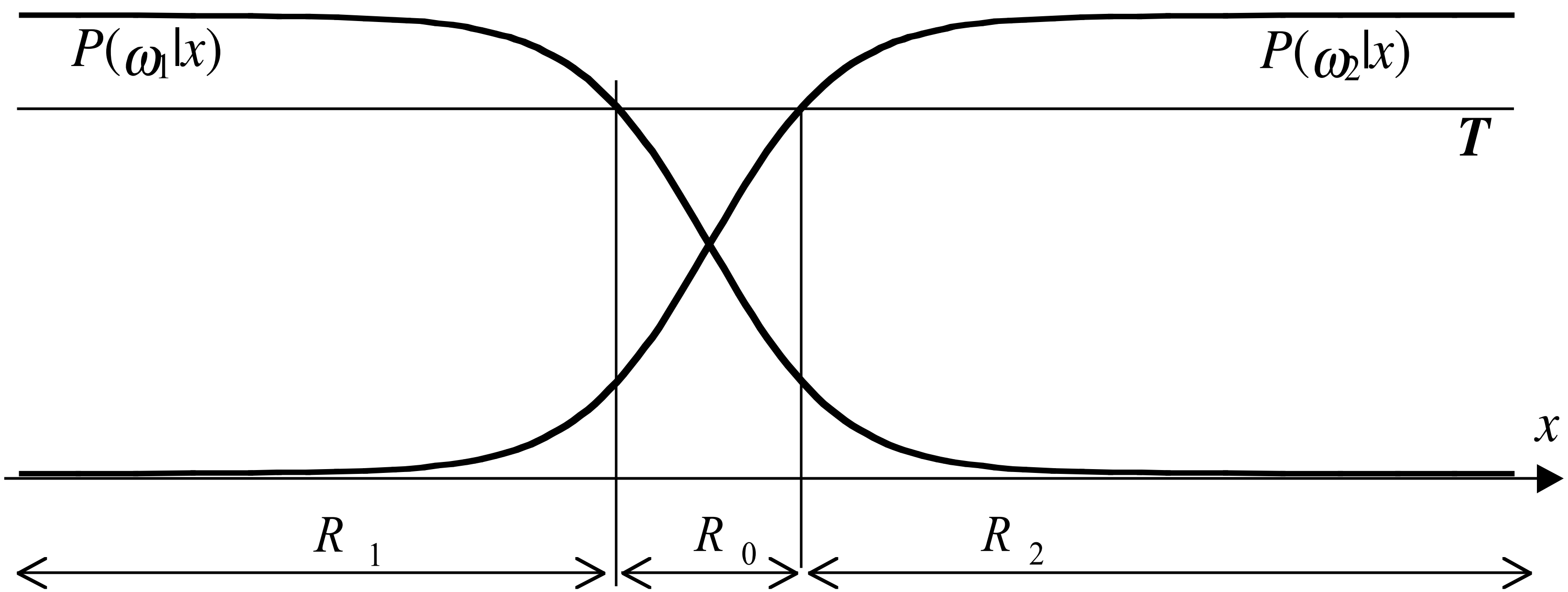
Illustration of the reject option. Inputs x such that the larger of the two posterior probabilities is less than or equal to some threshold θ will be rejected.

[C.Bishop, Pattern Recognition and Machine Learning, 2006]



- Classification errors arise from the regions of feature space where the largest of the posterior probabilities $p(\omega_k|x)$ is significantly less than unity, or equivalently where the joint distributions $p(x, \omega_k)$ have comparable values.
- These are the regions where we are relatively uncertain about class membership.
- In some applications, it will be appropriate to avoid making decisions on the difficult cases to reduce errors. This is known as **the reject option**.

Simple example of reject option



•Two classes with Gaussian distribution

•The reject threshold T identifies the reject region R_0

•This example clearly shows that error probability can be reduced by increasing the reject threshold T. Error becomes zero when the region R_0 contains all the patterns which are misclassified.

Error-reject trade-off and Chow's rule

Requirements of practical applications are often given as: *minimize error probability with P(reject) lower than "r"* (e.g., minimize error with reject<15%)

These requirements can be satisfied by the **Chow's rule** (C.K. Chow, 1957, 1970), that is the optimal rule with reject option:

$$\textit{if} \quad \max_i P(\omega_i / x) \geq T \rightarrow x \in \omega_i$$

$$\textit{otherwise} \text{ reject } x$$

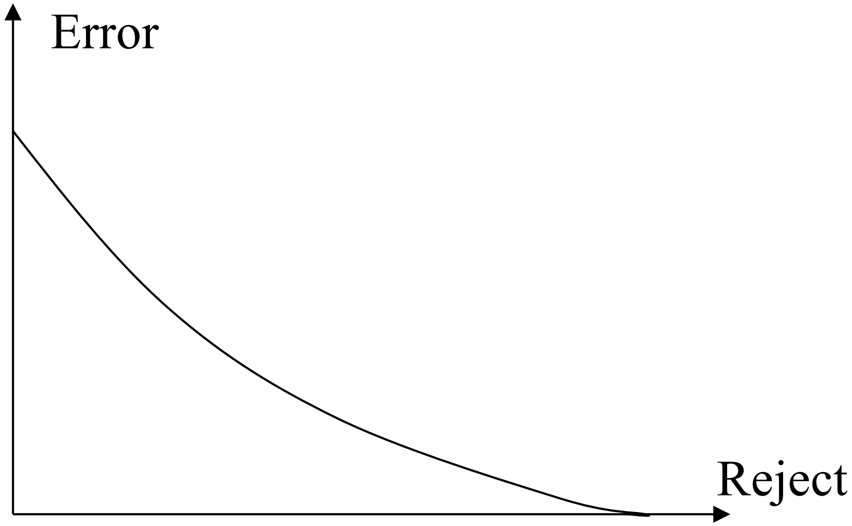
$$\textit{with } \mathrm{T} = \frac{\lambda_{\mathrm{e}} - \lambda_{\mathrm{r}}}{\lambda_{\mathrm{e}} - \lambda_{\mathrm{c}}}$$

- T is the **reject threshold**
- T ε [0..1] because $\lambda_c \leq \lambda_r$
- For T=0 ($\lambda_e=\lambda_r$) we have the classical MAP rule

➢ We can show (C.K. Chow, 1957) that Chow's rule minimizes error probability (that is, maximize classification accuracy) for any value of the reject probability.

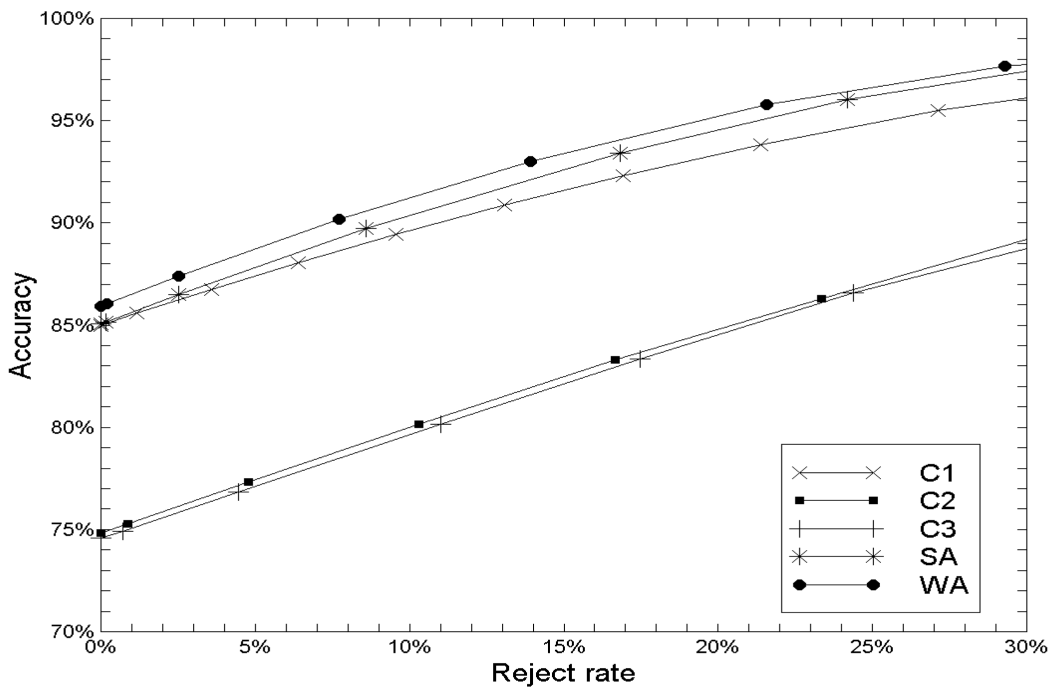
➢ It is easy to see that Chow's rule minimizes error by rejecting patterns for which the classification is not reliable enough.

Examples of error-reject trade-off



•Hypothetical trade-off curve

•*T* increases, then the rejection incresaes and error decreases (error-reject trade-off)



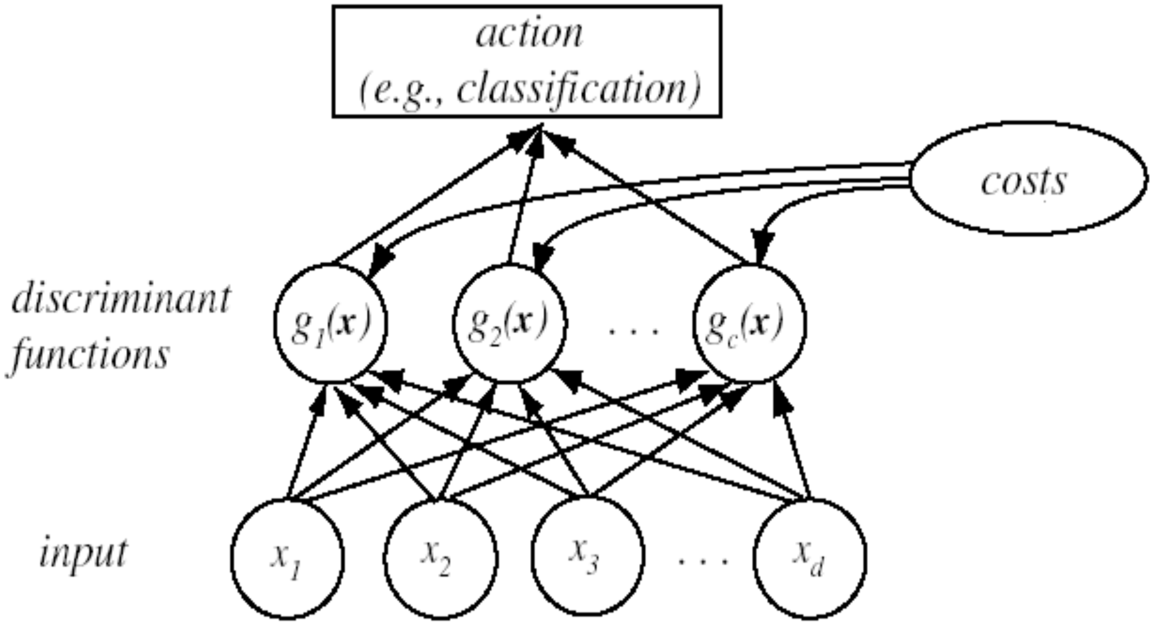
Examples of accuracy-rejection for different OCR (Optical Character Recognition) algorithms

Discriminant functions, decision surfaces/regions

So far we represented a classifier as a "machine" that takes as input the pattern **x** and assigns it to a class according to the minimu risk theory:

$$\mathbf{x} \longrightarrow \boxed{\mathbf{x} \rightarrow \omega_i \Leftrightarrow R(\omega_i \mid \mathbf{x}) < R(\omega_j \mid \mathbf{x}) \ \forall i \neq j, \ i=1,\ldots,c} \longrightarrow \boldsymbol{\omega}_{\mathbf{i}}$$

- An alternative representation of a pattern classifier is in terms of a set of ***discriminant functions*** $g_i(\mathbf{x})$, i=1,..,c.
- We assign **x** to the class ω_i if $g_i(\mathbf{x}) > g_j(\mathbf{x})$, $j \neq i$



Discriminant functions, decision surfaces/regions

- In general, we can consider $g_i(\mathbf{x})=-R(\omega_i|\mathbf{x})$; the discriminat function is aimed to minimize the risk.
- If we want to minimize the error probability: $g_i(\mathbf{x})=P(\omega_1/x)$
- We have many possible choices for $g_i(\mathbf{x})$; we can replace $g_i(\mathbf{x})$ with $f(g_i(\mathbf{x}))$, where f( ) is a increasing monotonic function.
- In particolar, if we want to minimize the error probability all the following choices are appropriate:

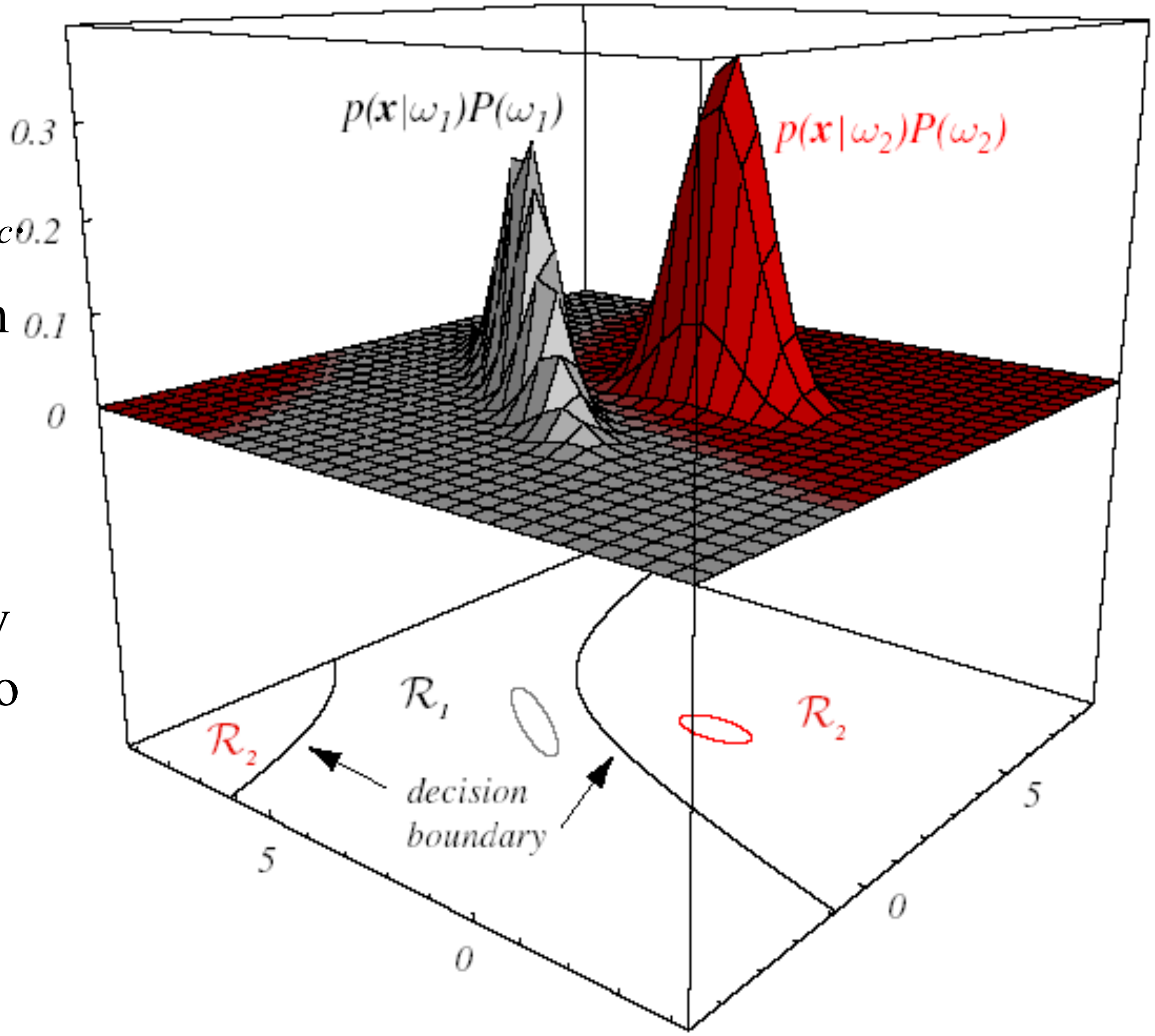
$$g_i(\mathbf{x}) = P(\omega_i/\mathbf{x}) = \frac{p(\mathbf{x}/\omega_i)P(\omega_i)}{\sum_{j=1}^{c} p(\mathbf{x}/\omega_j)P(\omega_j)}$$

$$g_i(\mathbf{x}) = p(\mathbf{x}/\omega_i)P(\omega_i)$$

$$g_i(\mathbf{x}) = \ln\left(p(\mathbf{x}/\omega_i)\right) + \ln\left(P(\omega_i)\right)$$

Discriminant functions, decision surfaces/regions

- Discriminat functions subdivide the "feature space" into *c decision regions* $R_1 \ldots R_c$
- If $g_i(\mathbf{x}) > g_j(\mathbf{x})$ for any $j \neq i$, then $\mathbf{x}\ \boldsymbol{\varepsilon}\ R_i$, and it is assigned to the class ω_i.
- **"Decision boundaries"** among regions are specified by $g_i(\mathbf{x}) = g_j(\mathbf{x})$, considering the two discriminant functions exhibiting maximum values



- Bi-dimensional example. Two classes with Gaussian distributions. Quadratic decision surfaces. Region R_2 is not simply connected.

Discriminant functions for binary classification

- For a two-class problem, we can use a single discriminant function $g(\mathbf{x})=g_1(\mathbf{x})- g_2(\mathbf{x})$
- Se $g(\mathbf{x})>0$ then ω_1, otherwise ω_2
- The following forms of the discriminant function can be used for a two-class problem:

$$g(\mathbf{x}) = P(\omega_1/\mathbf{x}) - P(\omega_2/\mathbf{x})$$

$$g(\mathbf{x}) = \ln\left(\frac{p(\mathbf{x}/\omega_1)}{p(\mathbf{x}/\omega_2)}\right) + \ln\left(\frac{P(\omega_1)}{P(\omega_2)}\right)$$

- A pattern classifier based on one of the above discriminat functions for a two-class task is called *dicotomic classifier.*

References

- Sections 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, Pattern Classification, R.O. Duda, P. E. Hart, and D. G. Stork, John Wiley & Sons, 2000
- Chapter 1, Statistical Pattern Recognition, Andrew Webb, John Wiley & Sons, 2002
- C.K. Chow, On optimum error and reject trade-off, IEEE Trans. on Information Theory 16 (1970) 41-46